

IBM Cognos Framework Manager  
Version 10.2.2

*Guide d'utilisation*



**Note**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 673.

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.ibm.com/ca/fr> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
Direction Qualité  
17, avenue de l'Europe  
92275 Bois-Colombes Cedex*

Le présent document s'applique à IBM Cognos Business Intelligence version 10.2.2 et peut aussi s'appliquer aux éditions ultérieures de ce produit.

© Copyright IBM Corporation 2005, 2014.

---

# Table des matières

<b>Avis aux lecteurs canadiens . . . . .</b>	<b>xxiii</b>
<b>Introduction . . . . .</b>	<b>xxv</b>
<b>Chapitre 1. Nouveautés . . . . .</b>	<b>1</b>
Nouveautés de la version 10.2.2 . . . . .	1
Nouvelles fonctions de la version 10.2.1 . . . . .	1
Nouvelles fonctions dans la version 10.2 . . . . .	1
Fonctions modifiées dans la version 10.2 . . . . .	2
<b>Chapitre 2. Initiation à l'utilisation de Framework Manager . . . . .</b>	<b>3</b>
Analyse du problème . . . . .	3
Création d'applications IBM Cognos Business Intelligence . . . . .	4
Objets que vous utiliserez . . . . .	5
Création de projets . . . . .	9
Ouverture de projets . . . . .	10
Page de projet . . . . .	10
Afficheur de projet . . . . .	11
Modification d'options pour des projets . . . . .	13
Réorganisation d'objets . . . . .	14
Onglet Explorateur . . . . .	15
Onglet Diagramme . . . . .	15
Onglet Grille des dimensions . . . . .	17
Sous-fenêtre Propriétés . . . . .	17
Sous-fenêtre Outils . . . . .	19
Conventions de dénomination pour les objets d'un projet . . . . .	21
Exemples de modèle . . . . .	22
Modèle d'entrepôt de données Vacances et Aventure. . . . .	23
Modèle Ventes Vacances et Aventure . . . . .	24
<b>Chapitre 3. Model Design Accelerator . . . . .</b>	<b>27</b>
Arborescence d'explorateur . . . . .	27
Diagramme d'explorateur. . . . .	28
Espace de travail Model Accelerator . . . . .	29
Diagramme de sujet de requête. . . . .	29
Utilisation du mode d'édition de relation . . . . .	30
Vue Avertissement de modèle . . . . .	31
Modification des paramètres de diagrammes . . . . .	31
Création d'un projet . . . . .	32
Schéma en étoile. . . . .	33
Gestion du schéma en étoile. . . . .	35
<b>Chapitre 4. Importation de métadonnées à partir de sources de données . . . . .</b>	<b>37</b>
Sources de données. . . . .	37
Sécurité des sources de données . . . . .	38
Types de connexions de source de données. . . . .	38
Métadonnées natives . . . . .	39
Connexions de source de données. . . . .	39
Création de connexions de sources de données . . . . .	43
Procédure de création d'une connexion de source de données . . . . .	45
Importation de métadonnées . . . . .	46
Importation de métadonnées à partir de bases de données relationnelles . . . . .	46
Importation de métadonnées à partir de modèles IBM Cognos . . . . .	48
Importation de métadonnées à partir de modèles Architect ou de catalogues Impromptu . . . . .	49

Importation de métadonnées à partir d'IBM Cognos DecisionStream ou d'IBM Cognos Data Manager . . . . .	49
Importation à partir de sources de métadonnées IBM . . . . .	53
Importation de métadonnées à partir de sources de métadonnées tiers . . . . .	57
Traitement des incidents liés aux métadonnées issues d'autres sources. . . . .	63
Importation de métadonnées en utilisant un fichier XML comme source de données . . . . .	64
Importation d'objets portant le même nom . . . . .	65
<b>Chapitre 5. Modélisation des métadonnées relationnelles . . . . .</b>	<b>67</b>
Configuration de la vue d'importation . . . . .	67
Configuration de la vue métier . . . . .	67
Relations . . . . .	68
Cardinalité . . . . .	69
Modification d'une relation . . . . .	73
Création d'expressions complexes pour une relation . . . . .	73
Création de relations . . . . .	73
Création d'un raccourci de relation . . . . .	74
Détection et génération de relations . . . . .	74
Sujets de requête . . . . .	76
Sujets de requête de source de données . . . . .	76
Sujets de requête de modèle . . . . .	78
Sujets de requête de procédure stockée . . . . .	80
Déterminants . . . . .	84
Création de sujets de requête de modèle basés sur des objets existants. . . . .	88
Affichage d'objets associés . . . . .	89
Création d'ensembles de requêtes . . . . .	89
Test de sujets de requête ou d'ensembles de requêtes . . . . .	93
Validation de sujets de requête . . . . .	96
Mise à jour de sujets de requête . . . . .	97
Conversion de sujets de requête en dimensions . . . . .	97
Conversion de sujets de requête de modèle en sujets de requête de source de données . . . . .	98
Edition du code SQL . . . . .	99
Modification du type de code SQL . . . . .	99
Modification du mode de génération du code SQL . . . . .	105
Dimensions . . . . .	106
Sources de données normalisées . . . . .	107
Création de dimensions ordinaires . . . . .	108
Tri des membres d'un niveau . . . . .	114
Rôles . . . . .	116
Création de dimensions de mesure . . . . .	117
Conversion de mesures en éléments de requête . . . . .	119
Relations de portée . . . . .	119
Création de dimensions ordinaires basées sur des objets existants . . . . .	121
Affichage d'objets associés . . . . .	121
Test de dimensions . . . . .	121
Conversion de dimensions ordinaires en sujets de requête . . . . .	124
Métadonnées multilingues . . . . .	125
Prise en charge des métadonnées multilingues . . . . .	125
Configuration d'un environnement de génération de rapports multilingue . . . . .	126
Modélisation avec des sources de données multilingues . . . . .	127
Utilisation de macros pour modéliser des données multilingues . . . . .	128
Ajout d'une langue à un projet . . . . .	129
Exportation des tables de traduction. . . . .	130
Importation des tables de traduction . . . . .	130
Exemple - Création de projets multilingues pour des métadonnées relationnelles . . . . .	131
Eléments de requête . . . . .	132
Modification du mode d'agrégation des éléments de requête . . . . .	136
Formatage des éléments de requête . . . . .	145
Définition des contrôles d'invite . . . . .	145
Conversion d'éléments de requête en mesures . . . . .	151
Règles métier . . . . .	151
Création de calculs . . . . .	152

Création de filtres . . . . .	155
Application de filtres . . . . .	157
Exemple - Affichage du nom de la devise pour chaque pays . . . . .	159
Création de tables de mappage des paramètres . . . . .	160
Exemple - Spécification de valeurs de langue pour des métadonnées relationnelles . . . . .	162
Création de paramètres de session . . . . .	163
Utilisation de paramètres avec des sujets de requête de source de données relationnelles . . . . .	165
Création d'invites avec des macros de requête . . . . .	165
Organisation des modèles . . . . .	176
Création de groupes sous forme de schémas en étoile . . . . .	177
Raccourcis . . . . .	181
Création de dossiers ou d'espaces-noms . . . . .	185
Création de dossiers d'élément de requête . . . . .	186
Création de dossiers de mesures . . . . .	187
Modèles durables . . . . .	187
Création d'un modèle durable . . . . .	188
Test d'un modèle durable . . . . .	189
Analyse de modèles . . . . .	190
Procédure d'analyse d'un modèle . . . . .	195
<b>Chapitre 6. Métadonnées SAP BW . . . . .</b>	<b>197</b>
Importation à partir d'une source de données SAP BW . . . . .	197
Accès aux métadonnées et données SAP BW . . . . .	198
Structures SAP BW . . . . .	199
Hiérarchies SAP BW . . . . .	200
Procédure d'accès à un InfoCube sécurisé . . . . .	202
Procédure d'importation à partir d'une source de données SAP BW . . . . .	202
Mappage d'objets SAP BW sur des objets Framework Manager . . . . .	203
Dimensions (SAP BW) . . . . .	204
Modification d'une dimension ordinaire (SAP BW) . . . . .	205
Rôles (SAP BW) . . . . .	211
Modification d'une dimension de chiffres clés (SAP BW) . . . . .	213
Affichage d'objets associés (SAP BW) . . . . .	214
Test d'une dimension ou d'un autre objet (SAP BW) . . . . .	215
Sujets de requête de modèle (SAP BW) . . . . .	218
Eléments de requête (SAP BW) . . . . .	220
Modification du mode d'agrégation des éléments de requête (SAP BW) . . . . .	224
Formatage des éléments de requête (SAP BW) . . . . .	227
Définition d'un contrôle d'invite (SAP BW) . . . . .	228
Variables SAP BW . . . . .	233
Valeurs de propriété de variable numérique (SAP BW) . . . . .	237
Valeurs de propriété de variable de caractéristique (SAP BW) . . . . .	238
Invites de sélection (SAP BW) . . . . .	239
Ajout de règles métier (SAP BW) . . . . .	239
Création d'un calcul (SAP BW) . . . . .	240
Création d'un filtre (SAP BW) . . . . .	242
Application d'un filtre (SAP BW) . . . . .	244
Création d'une table de mappage des paramètres (SAP BW) . . . . .	246
Création d'un paramètre de session (SAP BW) . . . . .	248
Organisation du modèle (SAP BW) . . . . .	250
Raccourcis (SAP BW) . . . . .	251
Création de dossiers ou d'espaces-noms (SAP BW) . . . . .	251
<b>Chapitre 7. Publication de packs . . . . .</b>	<b>255</b>
Vérification des modèles ou des packs . . . . .	255
Création ou modification de packs . . . . .	257
Création d'un pack . . . . .	259
Modification d'un pack . . . . .	259
Sécurité . . . . .	260
Utilisateurs, groupes et rôles . . . . .	260

Ajout de sécurité des données . . . . .	261
Sécurité des objets . . . . .	263
Modification de la sécurité d'un pack . . . . .	265
Spécification de langues . . . . .	266
Spécification de langues pour un pack . . . . .	266
Spécification de langues pour tous les packs . . . . .	267
Définition d'options de suppression . . . . .	267
Dimensions et sujets de requête externalisés . . . . .	267
Publication de packs . . . . .	269
Publication de packs basés sur des sources de données OLAP . . . . .	271
Publication de packs en exécutant des scripts. . . . .	273
Mise à jour de rapports pour utiliser la version la plus récente de packs . . . . .	274

## **Chapitre 8. Gestion de projet. . . . . 275**

Métadonnées de votre modèle . . . . .	275
Exploration de packs . . . . .	275
Affichage de la distribution d'objets dans des packs . . . . .	276
Création de la documentation du modèle . . . . .	276
Modélisation multiutilisateur . . . . .	277
Branchement et fusion de projets . . . . .	277
Projets segmentés et liés . . . . .	283
Contrôle de référentiel externe. . . . .	288
Administration des métadonnées . . . . .	288
Copie, déplacement, changement de nom ou suppression de projets . . . . .	289
Analyse de l'impact des modifications apportées à des packs . . . . .	291
Remappage d'objets à de nouvelles sources . . . . .	293
Exportation de métadonnées . . . . .	295
Réutilisation d'un projet . . . . .	296
Portabilité des modèles . . . . .	298
Synchronisation des projets. . . . .	305
Procédure de synchronisation . . . . .	307
Comportement des requêtes . . . . .	308
Restrictions . . . . .	309
Spécification de l'emplacement du traitement des cumuls d'agrégat . . . . .	316
Optimisation des performances en configurant le type de traitement de requête . . . . .	317
Amélioration des performances grâce à la réutilisation des données mises en cache lors de l'exécution d'un rapport . . . . .	318
Définition des modes de transaction. . . . .	320
Sélection d'ensembles de fonctions . . . . .	321
Qualité de service . . . . .	321
Contrôle et optimisation des requêtes . . . . .	324

## **Chapitre 9. Guide de modélisation des métadonnées . . . . . 327**

Présentation des concepts de modélisation d'IBM Cognos . . . . .	327
Concepts de modélisation relationnelle . . . . .	327
Considérations sur la conception de modèles. . . . .	338
Concepts de modélisation dimensionnelle . . . . .	346
Construction du modèle relationnel . . . . .	348
Définition de la base de la modélisation relationnelle . . . . .	349
Définition de la représentation dimensionnelle du modèle . . . . .	358
Organisation du modèle. . . . .	361

## **Chapitre 10. Code SQL généré par IBM Cognos Software . . . . . 365**

Présentation des requêtes dimensionnelles. . . . .	365
Requête à fait unique. . . . .	365
Requête à granularité multiple et à faits multiples sur des dimensions conformes . . . . .	367
Modélisation de relations de type 1-n en relations de type 1-1 . . . . .	370
Requête à granularité multiple et à faits multiples sur des dimensions non conformes . . . . .	371
Résolutions des dimensions et faits identifiés de façon ambiguë . . . . .	376
Sujets de requête représentant un niveau de hiérarchie . . . . .	376

Résolution de requêtes fractionnées par erreur . . . . .	377
<b>Chapitre 11. Mise à niveau de modèles . . . . .</b>	<b>383</b>
Vérification des modèles avant la mise à niveau . . . . .	383
Ouverture et la mise à niveau des modèles . . . . .	383
Mise à niveau et restrictions . . . . .	384
Mise à niveau et types de données . . . . .	385
Mise à niveau et sujets de requêtes basés sur des métadonnées SAP BW . . . . .	385
Vérification et réparation de modèles IBM Cognos ReportNet mis à niveau . . . . .	386
Conversion d'informations de dimension en déterminants ou en dimensions . . . . .	386
Sélection et réparation d'objets dans les modèles IBM Cognos ReportNet mis à niveau . . . . .	390
Mise à niveau de projets segmentés et liés . . . . .	391
<b>Annexe A. Fonctions d'accessibilité. . . . .</b>	<b>393</b>
Fonctions d'accessibilité dans Framework Manager . . . . .	393
Raccourcis-clavier pour Framework Manager . . . . .	393
Raccourcis-clavier pour Model Design Accelerator . . . . .	393
IBM et l'accessibilité . . . . .	394
<b>Annexe B. Traitement des incidents. . . . .</b>	<b>395</b>
Impossible de comparer deux objets CLOB dans Oracle . . . . .	395
Erreur de mémoire insuffisante avec des métadonnées importées ERWin . . . . .	395
Framework Manager ne peut pas accéder à l'URI de la passerelle . . . . .	395
Les noms d'objets s'affichent dans une langue erronée . . . . .	396
Dans Oracle, des jointures externes complètes renvoient des résultats incorrects . . . . .	396
Erreur lors du test des sujets de requête dans un modèle importé à partir de Teradata . . . . .	396
Erreur liée au sujet de requête SQL en saisie directe . . . . .	397
Erreur QE-DEF-0259 . . . . .	397
La dimension des données clés externalisées conserve une ancienne valeur d'invite . . . . .	398
Des modèles anciens affichent un niveau de sécurité pour des objets . . . . .	398
Echec de l'exportation d'un modèle Framework Manager vers un fichier CWM avec l'erreur Impossible de trouver MILOG.TXT . . . . .	398
Différence de code SQL pour les jointures internes après une mise à niveau vers IBM Cognos BI version 8.3 et versions ultérieures . . . . .	399
Jointures externes complètes non envoyées à Oracle 9i et 10GR1 . . . . .	399
Différences inexplicables dans les calculs de nombres . . . . .	399
Recherche de valeurs pouvant renvoyer des résultats inattendus . . . . .	401
<b>Annexe C. Utilisation de l'éditeur d'expression . . . . .</b>	<b>403</b>
Opérateurs . . . . .	403
( . . . . .	403
) . . . . .	403
* . . . . .	403
/ . . . . .	403
. . . . .	403
+ . . . . .	404
- . . . . .	404
< . . . . .	404
<= . . . . .	404
<> . . . . .	404
= . . . . .	404
> . . . . .	404
-> . . . . .	405
>= . . . . .	405
et . . . . .	405
auto . . . . .	405
between . . . . .	405
case . . . . .	405
contains . . . . .	406

currentMeasure.	406
default	406
distinct	406
else.	406
end.	406
ends with	407
escape.	407
for	407
for all	407
for any	407
for report.	408
if	408
in	408
in_range	408
is missing	409
is null	409
is not missing	409
is not null	409
like.	409
lookup	410
not	410
or	410
prefilter	410
rows	411
rows	411
starts with	411
then	411
when	411
Récapitulatifs	411
Fonctions statistiques.	411
aggregate.	420
average	420
count	421
maximum	421
median	421
minimum.	422
moving-average	422
moving-total.	422
percentage	423
percentile.	424
quantile	424
quartile	425
rank	425
running-average	426
running-count	427
running-difference.	427
running-maximum	428
running-minimum.	428
running-total	429
total	429
Récapitulatifs des membres.	430
aggregate.	430
average	430
count	430
maximum	431
median	431
minimum.	431
percentage	432
percentile.	432
quantile	432
quartile	432



rank	432
standard-deviation	433
standard-deviation-pop	433
total	433
variance	433
variance-pop	433
Constantes	433
date	433
date-time	434
Heure avec fuseau horaire (time with time zone)	434
Horodatage avec fuseau horaire (timestamp with time zone)	434
interval	434
Intervalle de type Année (interval year)	434
Intervalle de type Mois (interval month)	434
Intervalle de type Année à mois (interval year to month)	434
Intervalle de type Jour (interval day)	434
Intervalle de type Heure (interval hour)	434
Intervalle de type Minute (interval minute)	434
Intervalle de type Seconde (interval second)	434
Intervalle de type Jour à heure (interval day to hour)	434
Intervalle de type Jour à minute (interval day to minute)	434
Intervalle de type Jour à seconde (interval day to second)	434
Intervalle de type Heure à minute (interval hour to minute)	434
Intervalle de type Heure à seconde (interval hour to second)	435
Intervalle de type minute à seconde (interval minute to second)	435
null	435
number	435
string	435
time	435
Syntaxes	435
if then else	435
in_range	435
CASE (recherche)	436
CASE (simple)	436
Fonctions de date-heure selon le principe des jours ouvrables	436
_add_seconds	436
_add_minutes	437
_add_hours	438
_add_days	438
_add_months	439
_add_years	440
_age	440
_day_of_week	440
_day_of_year	441
_days_between	441
_days_to_end_of_month	441
_first_of_month	442
_last_of_month	442
_make_timestamp	442
_months_between	443
_shift_timezone	443
_week_of_year	444
_years_between	445
_ymdint_between	445
Fonctions de bloc	445
_firstFromSet	445
_remainderSet	446
Aster Data	447
Chaîne d'Aster DATA	447
Formatage du type de données Aster DATA	448
Mathématiques appliquées à Aster DATA	449

Trigonométrie sur Aster DATA . . . . .	449
ascii . . . . .	451
chr . . . . .	451
current_schema . . . . .	451
translate . . . . .	451
date_trunc . . . . .	451
version . . . . .	451
DB2 . . . . .	451
Fonctions mathématiques DB2 . . . . .	451
Fonctions trigonométriques DB2 . . . . .	452
ascii . . . . .	454
ceiling . . . . .	454
char . . . . .	454
chr . . . . .	454
concat . . . . .	454
date . . . . .	455
day . . . . .	455
dayname . . . . .	455
dayofweek . . . . .	456
dayofweek_iso . . . . .	456
dayofyear . . . . .	456
days . . . . .	457
dec . . . . .	457
decimal . . . . .	457
difference . . . . .	457
digits . . . . .	458
double . . . . .	458
event_mon_state . . . . .	458
float . . . . .	458
hex . . . . .	458
hour . . . . .	458
insert . . . . .	459
integer . . . . .	459
int . . . . .	459
julian_day . . . . .	459
lcase . . . . .	460
left . . . . .	460
length . . . . .	460
locate . . . . .	460
long_varchar . . . . .	461
ltrim . . . . .	461
microsecond . . . . .	461
midnight_seconds . . . . .	461
minute . . . . .	462
month . . . . .	462
monthname . . . . .	462
quarter . . . . .	463
radians . . . . .	463
repeat . . . . .	463
replace . . . . .	463
right . . . . .	464
round . . . . .	464
rtrim . . . . .	464
second . . . . .	464
sign . . . . .	465
smallint . . . . .	465
soundex . . . . .	465
space . . . . .	465
substr . . . . .	466
table_name . . . . .	466
table_schema . . . . .	466

time	466
timestamp	466
timestamp_iso	467
timestampdiff	467
to_char	467
to_date	467
translate	467
trunc	468
truncate	468
ucase	469
value	469
varchar	469
week	469
year	469
Greenplum	470
Chaîne Greenplum	470
Formatage du type de données Greenplum	472
Fonctions mathématiques Greenplum	472
Fonctions trigonométriques Greenplum.	473
ascii	474
chr	474
current_database	474
current_schema	474
{current_user}	474
{session_user}	475
translate	475
date_trunc	475
version	475
Informix	475
Fonctions mathématiques Informix	475
Fonctions trigonométriques Informix	476
cardinality	476
char_length	477
concat	477
date	477
day	477
extend	477
hex	478
initcap	478
length	478
lpad	478
mdy	478
month	478
nvl	479
octet_length	479
replace	479
round	479
rpad	480
substr	480
to_char	480
to_date	480
trunc	481
weekday	481
year	481
MS Access	481
Fonctions de conversion MS Access	481
Fonctions mathématiques MS Access	482
Fonctions trigonométriques MS Access	482
ascii	483
ceiling	483
chr	483

concat . . . . .	483
curdate . . . . .	484
curtime . . . . .	484
dayname . . . . .	484
dayofmonth . . . . .	484
dayofweek . . . . .	484
dayofyear . . . . .	484
hour . . . . .	484
instr . . . . .	485
lcase . . . . .	485
left . . . . .	485
length . . . . .	485
locate . . . . .	485
ltrim . . . . .	485
minute . . . . .	486
month . . . . .	486
monthname . . . . .	486
now . . . . .	486
position . . . . .	486
quarter . . . . .	486
right . . . . .	487
round . . . . .	487
rtrim . . . . .	487
sign . . . . .	487
space . . . . .	487
substr . . . . .	488
sous-chaine . . . . .	488
truncate . . . . .	488
ucase . . . . .	488
week . . . . .	488
year . . . . .	489
MySQL . . . . .	489
Chaîne MySQL . . . . .	489
Fonctions mathématiques MySQL . . . . .	490
Fonctions trigonométriques MySQL . . . . .	491
ascii . . . . .	492
database . . . . .	492
schema . . . . .	492
session_user . . . . .	492
system_user . . . . .	492
version . . . . .	492
Netezza . . . . .	493
Fonctions mathématiques Netezza . . . . .	493
Fonctions trigonométriques Netezza . . . . .	493
Netezza Fuzzy . . . . .	494
Fonctions phonétiques Netezza . . . . .	494
ascii . . . . .	495
chr . . . . .	495
decode . . . . .	495
initcap . . . . .	496
instr . . . . .	496
lpad . . . . .	496
ltrim . . . . .	496
months_between . . . . .	496
next_day . . . . .	496
nvl . . . . .	497
round . . . . .	497
rpad . . . . .	497
rtrim . . . . .	497
substr . . . . .	497
{current_db} . . . . .	498

{current_user}	498
{session_user}	498
to_char	498
to_date	498
to_number	498
translate	498
date_trunc	499
trunc	499
version	499
Oracle	499
Fonctions mathématiques Oracle	499
Fonctions trigonométriques Oracle	499
add_months	501
ascii	501
ceil	501
char_length	501
chr	501
concat	501
decode	502
dump	502
greatest	502
initcap	502
instr	502
instrb	503
least	503
length	503
lengthb	503
lpad	503
ltrim	503
months_between	504
new_time	504
next_day	504
nls_initcap	504
nls_lower	504
nls_upper	504
nvl	505
replace	505
round	505
rpad	505
rtrim	506
sign	506
soundex	506
substr	506
substrb	506
{sysdate}	507
to_char	507
to_date	507
to_number	507
translate	508
trunc	508
trunc	508
{user}	508
vsize	508
Paracel	509
Chaîne Paracel	509
Formatage des types de données Paracel	509
Fonctions mathématiques Paracel	510
current_database	510
current_schema	510
{current_user}	510
{session_user}	510

translate . . . . .	510
version . . . . .	510
Postgres . . . . .	511
Chaîne Postgres . . . . .	511
Formatage des types de données Postgres. . . . .	512
Fonctions mathématiques Postgres . . . . .	513
Fonctions trigonométriques Postgres. . . . .	514
ascii . . . . .	515
chr . . . . .	515
{current_db} . . . . .	515
{current_catalog} . . . . .	515
{current_schema} . . . . .	515
{current_user} . . . . .	515
{session_user} . . . . .	516
translate . . . . .	516
date_trunc . . . . .	516
version . . . . .	516
Red Brick. . . . .	516
ceil . . . . .	516
concat . . . . .	516
{current_user} . . . . .	517
date . . . . .	517
dateadd . . . . .	517
datediff . . . . .	517
datetime. . . . .	517
dec . . . . .	517
decimal . . . . .	518
decode . . . . .	518
float . . . . .	518
ifnull . . . . .	518
int . . . . .	518
integer . . . . .	518
length . . . . .	519
lengthb . . . . .	519
ltrim . . . . .	519
nullif . . . . .	519
positionb . . . . .	519
real. . . . .	519
round . . . . .	519
rtrim . . . . .	520
sign . . . . .	520
string . . . . .	520
substr . . . . .	520
substrb . . . . .	521
time . . . . .	521
timestamp . . . . .	521
timestamp . . . . .	521
to_char . . . . .	521
SAP BW . . . . .	521
Fonctions trigonométriques SAP BW . . . . .	521
Fonctions mathématiques SAP BW . . . . .	523
Salesforce.com . . . . .	523
Fonctions de date . . . . .	523
convertCurrency . . . . .	526
convertTimezone . . . . .	526
SQL Server . . . . .	527
Fonctions mathématiques SQL Server . . . . .	527
Fonctions trigonométriques SQL Server. . . . .	527
ascii . . . . .	528
char . . . . .	529
charindex. . . . .	529

{current_user}	529
datalength	529
dateadd	529
datediff	530
datetime	530
datepart	530
day	530
difference	531
getdate	531
isnull	531
left	531
ltrim	532
month	532
patindex	532
replace	532
replicate	532
inverse	533
right	533
round	533
rtrim	533
sign	533
soundex	533
space	534
str	534
stuff	534
year	534
Sybase	534
Fonctions mathématiques Sybase	534
Fonctions trigonométriques Sybase	535
ascii	536
char	536
charindex	537
datalength	537
dateadd	537
datediff	537
datetime	537
datepart	538
jour	538
difference	538
getdate	539
left	539
ltrim	539
month	539
patindex	539
rand	540
replicate	540
inverse	540
right	540
round	540
rtrim	540
soundex	541
space	541
str	541
stuff	541
substring	541
to_unichar	542
uhighsurr	542
ulowsurr	542
uscalar	542
year	543
Teradata	543

Fonctions trigonométriques Teradata	543
account	544
add_months	545
bytes	545
case_n	545
char2hexint	545
characters	545
database	545
date	546
format	546
index	546
log	546
nullif	546
nullifzero	546
profile	546
random	547
role	547
session	547
soundex	547
substr	547
time	547
type	548
user	548
vargraphic	548
zeroifnull	548
Vectorwise	548
Chaîne Vectorwise	548
Fonctions mathématiques Vectorwise	549
Fonctions trigonométriques Vectorwise	549
{initial_user}	550
{session_user}	550
numeric_trunc	551
Vertica	551
Chaîne Vertica	551
Formatage des types de données Vertica	552
Fonctions mathématiques Vertica	553
Fonctions trigonométriques Vertica	553
ascii	555
chr	555
current_database	555
current_schema	555
{current_user}	555
{session_user}	555
translate	555
date_trunc	556
version	556
Fonctions macros	556
+	556
_add_seconds	556
_add_minutes	557
_add_hours	558
_add_days	558
_add_months	559
_add_years	559
array	560
csv	560
dq	560
getConfigurationEntry	561
grep	561
_first_of_month	561
_last_of_month	562



join . . . . .	562
lstrip . . . . .	562
prompt . . . . .	563
promptmany . . . . .	564
rstrip . . . . .	565
sb . . . . .	565
sq . . . . .	565
sort . . . . .	566
split . . . . .	566
strip . . . . .	566
substitute . . . . .	567
timestampMask . . . . .	567
toLocal . . . . .	568
tolower . . . . .	569
toupper . . . . .	569
toUTC . . . . .	570
uniques . . . . .	570
urlencode . . . . .	571
CSIdentityName . . . . .	571
CSIdentityNameList . . . . .	571
CAMPassport . . . . .	571
CAMIDList . . . . .	572
CAMIDListForType . . . . .	572
CASE (simple) . . . . .	572
Fonctions communes . . . . .	573
abs . . . . .	573
cast . . . . .	573
ceil . . . . .	574
ceiling . . . . .	574
char_length . . . . .	574
character_length . . . . .	575
coalesce . . . . .	575
current_date . . . . .	575
current_time . . . . .	575
current_timestamp . . . . .	576
exp . . . . .	576
extract . . . . .	576
floor . . . . .	577
ln . . . . .	577
localtime . . . . .	577
localtimestamp . . . . .	578
lower . . . . .	578
mod . . . . .	578
nullif . . . . .	578
octet_length . . . . .	578
position . . . . .	579
power . . . . .	579
_round . . . . .	580
row . . . . .	580
sqrt . . . . .	581
substring . . . . .	581
trim . . . . .	581
upper . . . . .	582
width-bucket . . . . .	582
Fonctions trigonométriques . . . . .	583
Fonctions dimensionnelles . . . . .	585
ancestor . . . . .	585
ancestors . . . . .	586
bottomCount . . . . .	587
bottomPercent . . . . .	587
bottomSum . . . . .	587

caption . . . . .	588
children . . . . .	588
closingPeriod . . . . .	589
cousin . . . . .	589
completeTuple . . . . .	590
currentMember . . . . .	591
defaultMember . . . . .	591
descendants . . . . .	591
except . . . . .	593
filter . . . . .	593
firstChild . . . . .	594
firstSibling . . . . .	594
_format . . . . .	594
emptySet . . . . .	595
generate . . . . .	595
head . . . . .	596
hierarchize . . . . .	596
hierarchy . . . . .	596
item . . . . .	597
intersect . . . . .	597
lag . . . . .	598
lastChild . . . . .	598
lastPeriods . . . . .	598
lastSibling . . . . .	599
lead . . . . .	599
level . . . . .	600
levels . . . . .	600
linkMember . . . . .	601
members . . . . .	601
nextMember . . . . .	601
openingPeriod . . . . .	602
order . . . . .	602
ordinal . . . . .	603
parallelPeriod . . . . .	604
parent . . . . .	604
periodsToDate . . . . .	604
prevMember . . . . .	605
member . . . . .	605
nestedSet . . . . .	606
set . . . . .	606
siblings . . . . .	606
tail . . . . .	607
topCount . . . . .	607
topPercent . . . . .	608
topSum . . . . .	608
tuple . . . . .	608
union . . . . .	609
roleValue . . . . .	609
rootMember . . . . .	610
rootMembers . . . . .	610
subset . . . . .	610
uniques . . . . .	611
value . . . . .	611
Fonctions de rapport . . . . .	612
Today . . . . .	612
Now . . . . .	612
AsOfDate . . . . .	612
AsOfTime . . . . .	612
ReportDate . . . . .	612
ReportName . . . . .	612
ReportPath . . . . .	613

ReportDescription . . . . .	613
ReportLocale . . . . .	613
GetLocale . . . . .	613
Locale . . . . .	613
ReportProductLocale . . . . .	613
ReportAuthorLocale . . . . .	613
ReportSaveDate . . . . .	613
ReportCreateDate . . . . .	614
ReportID . . . . .	614
ReportOutput . . . . .	614
ReportOption . . . . .	614
ServerName . . . . .	614
ServerLocale. . . . .	614
ModelPath . . . . .	614
BurstKey . . . . .	615
BurstRecipients. . . . .	615
IsBursting . . . . .	615
ParamNames . . . . .	615
ParamName . . . . .	615
ParamDisplayValue . . . . .	615
ParamValue . . . . .	615
ParamCount. . . . .	615
RowNumber . . . . .	616
PageNumber . . . . .	616
PageCount . . . . .	616
IsPageCountAvailable . . . . .	616
HorizontalPageNumber . . . . .	616
HorizontalPageCount. . . . .	616
PageName . . . . .	616
URLEncode . . . . .	616
TOCHeadingCount . . . . .	617
IsAccessible . . . . .	617
IsBooklet . . . . .	617
ColumnNumber . . . . .	617
IsCrosstabRowNodeMember . . . . .	617
IsCrosstabColumnNodeMember . . . . .	617
IsInnerMostCrosstabRowNodeMember . . . . .	617
IsInnerMostCrosstabColumnNodeMember. . . . .	618
IsOuterMostCrosstabRowNodeMember. . . . .	618
IsOuterMostCrosstabColumnNodeMember . . . . .	618
IsFirstColumn . . . . .	618
IsLastColumn . . . . .	618
IsLastInnerMostCrosstabColumnNodeMember . . . . .	618
IsLastInnerMostCrosstabRowNodeMember . . . . .	618
CubeName . . . . .	619
CubeDescription . . . . .	619
CubeCreatedOn . . . . .	619
CubeDataUpdatedOn. . . . .	619
CubeSchemaUpdatedOn. . . . .	619
CubeIsOptimized . . . . .	619
CubeDefaultMeasure . . . . .	619
CubeCurrentPeriod . . . . .	620
CellValue. . . . .	620
InScope . . . . .	620
Fonctions de conversion du type de données . . . . .	620
_add_days . . . . .	620
_add_months . . . . .	620
_add_years . . . . .	620
_age . . . . .	621
_day_of_week . . . . .	621
_day_of_year . . . . .	621

_days_between . . . . .	621
_days_to_end_of_month . . . . .	622
_first_of_month . . . . .	622
_last_of_month . . . . .	622
_make_timestamp . . . . .	622
_months_between . . . . .	622
_week_of_year . . . . .	623
_years_between . . . . .	623
_ymdint_between . . . . .	623
abs . . . . .	623
partie entière par excès . . . . .	623
character_length . . . . .	623
date2string . . . . .	624
date2timestamp . . . . .	624
date2timestampTZ . . . . .	624
DTinterval2string . . . . .	624
DTinterval2stringAsTime . . . . .	624
exp . . . . .	624
extract . . . . .	625
partie-entière . . . . .	625
int2DTinterval . . . . .	625
int2YMininterval . . . . .	626
ln . . . . .	626
minuscules . . . . .	626
mapNumberToLetter . . . . .	626
mod . . . . .	626
nullif . . . . .	626
number2string . . . . .	627
octet_length . . . . .	627
position . . . . .	627
power . . . . .	627
round . . . . .	627
racine-carrée . . . . .	628
status . . . . .	628
string2date . . . . .	628
string2double . . . . .	628
string2DTinterval . . . . .	628
string2int32 . . . . .	628
string2int64 . . . . .	628
string2time . . . . .	629
string2timestamp . . . . .	629
string2timestampTZ . . . . .	629
string2YMininterval . . . . .	629
sous-chaîne . . . . .	629
time2string . . . . .	629
timestamp2date . . . . .	630
timestamp2string . . . . .	630
timestamp2timestampTZ . . . . .	630
timestampTZ2date . . . . .	630
timestampTZ2string . . . . .	630
timestampTZ2timestamp . . . . .	630
timeTZ2string . . . . .	630
trim . . . . .	631
majuscules . . . . .	631
YMininterval2string . . . . .	631

**Annexe D. Aide-mémoire sur le formatage de données . . . . . 633**

Propriétés de formatage de données . . . . .	633
Caractères 'non applicable' . . . . .	633
Caractères d'erreur (Tous) . . . . .	633
Type de calendrier . . . . .	633

Format de l'heure . . . . .	633
Devise. . . . .	633
Afficher la devise . . . . .	633
Symbole de la devise. . . . .	634
Position du symbole de la devise. . . . .	634
Ordre de la date . . . . .	634
Séparateur de date . . . . .	634
Style de date . . . . .	634
Séparateur de décimales. . . . .	634
Format des caractères numériques . . . . .	634
Format des caractères numériques . . . . .	634
Afficher les symboles AM / PM . . . . .	635
Afficher sous forme d'exposant . . . . .	635
Afficher les jours . . . . .	635
Afficher les périodes . . . . .	635
Afficher les heures . . . . .	635
Afficher les millisecondes . . . . .	635
Afficher les minutes . . . . .	635
Afficher les mois . . . . .	635
Afficher les mois . . . . .	635
Afficher les secondes . . . . .	636
Afficher les fuseaux horaires . . . . .	636
Afficher les jours de la semaine . . . . .	636
Afficher les années . . . . .	636
Afficher les années . . . . .	636
Caractères de division par zéro . . . . .	636
Symbole de l'exposant . . . . .	636
Taille du groupe (chiffres) . . . . .	636
Symbole international de la devise . . . . .	636
Mantisse (chiffres). . . . .	637
Nb. maximal de chiffres . . . . .	637
Nb. minimal de chiffres . . . . .	637
Caractères pour les valeurs manquantes . . . . .	637
Motif des nombres négatifs. . . . .	637
Position du signe négatif . . . . .	637
Signe négatif . . . . .	637
Nombre de décimales . . . . .	637
Caractères de dépassement numérique . . . . .	637
Caractère de cadrage . . . . .	637
Motifs . . . . .	638
Symbole de pourcentage. . . . .	638
Echelle des pourcentages (entier) . . . . .	638
Echelle . . . . .	638
Taille du groupe secondaire (chiffres) . . . . .	638
Caractères d'erreur de sécurité. . . . .	638
Séparateur des milliers . . . . .	638
Séparateur de l'heure. . . . .	638
Format de l'heure . . . . .	639
Unité de temps . . . . .	639
Utiliser le séparateur des milliers. . . . .	639
Caractères pour les valeurs nulles . . . . .	639
<b>Annexe E. Utilisation de motifs pour mettre en forme des données . . . . .</b>	<b>641</b>
Règles d'utilisation des motifs . . . . .	641
Symboles de date et d'heure . . . . .	642
Symboles des décimales . . . . .	650
<b>Annexe F. Guide de gestion des données SAP BW à des fins d'utilisation dans Transformer . . . . .</b>	<b>653</b>
Utilisation de données SAP BW via un pack dans Framework Manager . . . . .	653

Création d'une requête BW dans SAP Business Explorer Query Designer . . . . .	654
Création d'un pack dans Framework Manager . . . . .	657
Création d'un modèle dans Transformer . . . . .	659
Utilisation de données SAP BW à l'aide de fichiers CSV externalisés dans Framework Manager . . . . .	661
Exigences relatives à une requête SAP BW. . . . .	663
Considérations relatives à Framework Manager . . . . .	664
Création de PowerCubes à partir de données SAP BW . . . . .	666
<b>Annexe G. Mots réservés . . . . .</b>	<b>669</b>
<b>Annexe H. Types de données XML . . . . .</b>	<b>671</b>
<b>Remarques . . . . .</b>	<b>673</b>
<b>Glossaire . . . . .</b>	<b>677</b>
A . . . . .	677
C . . . . .	678
D . . . . .	679
E . . . . .	679
F . . . . .	680
G . . . . .	681
H . . . . .	681
I . . . . .	681
L . . . . .	681
M . . . . .	681
N . . . . .	682
O . . . . .	682
P . . . . .	682
R . . . . .	683
S . . . . .	684
T . . . . .	685
U . . . . .	685
V . . . . .	685
W . . . . .	685
Z . . . . .	685
<b>Index . . . . .</b>	<b>687</b>

---

## Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

### Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

### Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

### Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








### OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

### Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

## Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

## Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.



---

## Introduction

IBM® Cognos Framework Manager est un outil de modélisation des données. Un modèle est une présentation professionnelle des informations contenues dans une ou plusieurs sources de données. Lorsque vous ajoutez des fonctions de sécurité et des fonctions multilingues à cette présentation professionnelle, un modèle unique est capable de répondre aux besoins de nombreux groupes d'utilisateurs à travers le monde.

Ce document contient des procédures, des exemples, des remarques, des conseils et d'autres informations générales visant à faciliter la préparation d'un modèle pour la production de rapports et le déploiement d'un pack.

### Utilisateurs concernés

Ce document a pour but d'aider les modélisateurs de données à utiliser IBM Cognos Framework Manager. Avant d'utiliser Framework Manager, vous devez comprendre la modélisation de données et savoir écrire des requêtes.

### Recherche d'informations

Pour rechercher la documentation des produits sur le Web, y compris toutes les documentations traduites, accédez à la page IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

### Instructions prospectives

La présente documentation décrit les fonctionnalités actuelles du produit. Elle peut contenir des références à des éléments qui ne sont pas disponibles actuellement. Cela n'implique aucune disponibilité ultérieure de ces éléments. De telles références ne constituent en aucun cas un engagement, une promesse ou une obligation légale de fournir un élément, un code ou une fonctionnalité. Le développement, la disponibilité et le calendrier de mise à disposition des fonctions demeurent à la seule discrétion d'IBM.

### Clause de décharge relative aux exemples

La société Vacances et aventure, Ventes VA, toute variation du nom Vacances et aventure, ainsi que les exemples de planification, illustrent des opérations commerciales fictives, avec des exemples de données utilisées pour développer des exemples d'applications, destinées à l'usage d'IBM et de ses clients. Les exemples de données contenus dans ces enregistrements fictifs sont utilisés pour représenter des transactions de vente, la distribution de produits, des données financières et des ressources humaines. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite. D'autres fichiers d'exemple peuvent contenir des données fictives générées manuellement ou par une machine, des données factuelles compilées à partir de sources académiques ou publiques, ou des données utilisées avec l'autorisation du détenteur des droits d'auteur, à utiliser comme exemple de données pour développer des exemples d'application. Les noms de produit référencés peuvent être les marques de leurs propriétaires respectifs. Toute reproduction sans autorisation est interdite.

## **Fonctions d'accessibilité**

Les fonctions d'accessibilité permettent aux utilisateurs souffrant d'un handicap physique, comme une mobilité réduite ou une vision limitée, d'utiliser avec succès les produits informatiques. IBM Cognos Framework Manager est doté de fonctions d'accessibilité. Pour plus d'informations, voir Annexe A, «Fonctions d'accessibilité», à la page 393.

---

## Chapitre 1. Nouveautés

Cette section vous aidera à planifier les stratégies de mise à niveau et de déploiement, ainsi que les critères de formation requis pour votre application.

Pour des informations concernant la mise à niveau, consultez le manuel *IBM Cognos Business Intelligence - Guide d'installation et de configuration* et la rubrique Chapitre 11, «Mise à niveau de modèles», à la page 383.

Pour en savoir d'avantage sur les nouvelles fonctions de cette édition IBM Cognos Business Intelligence, voir le manuel *IBM Cognos Business Intelligence - Guide des nouveautés*.

Pour consulter la liste actualisée des environnements pris en charge par les produits IBM Cognos Business Intelligence, y compris des informations sur les systèmes d'exploitation, les correctifs, les navigateurs, les serveurs Web, les serveurs d'annuaire, les serveurs de base de données et les serveurs d'applications, reportez-vous à la page IBM Software Product Compatibility Reports ([www.ibm.com/support/docview.wss?uid=swg27042164](http://www.ibm.com/support/docview.wss?uid=swg27042164)).

---

### Nouveautés de la version 10.2.2

Cette édition d'IBM Cognos Framework Manager ne comporte aucune nouvelle fonction.

---

### Nouvelles fonctions de la version 10.2.1

Les fonctions suivantes ont été introduites dans cette édition d'IBM Cognos Framework Manager.

#### **Nouvelles propriétés pour contrôler les transactions en lecture/écriture**

Deux nouvelles propriétés sont disponibles pour les sources de données Framework Manager. Il s'agit de Mode d'accès de la transaction et de Mode de l'instruction de transaction.

Par défaut, un sujet de requête qui contient une procédure stockée est exécuté en tant que transaction en lecture seule. Cependant, la procédure stockée peut inclure des opérations qui nécessitent une transaction de type lecture/écriture. Dans les éditions précédentes de Framework Manager, les sujets de requête qui utilisaient des procédures pouvaient être exécutés dans une transaction en lecture/écriture. Cependant, ils n'étaient visibles par les auteurs que dans Event Studio. Les nouvelles propriétés permettent l'utilisation dans tous les autres studios des sujets de requête exécutés dans des transactions de type lecture/écriture.

Pour plus d'informations, voir «Définition des modes de transaction», à la page 320

---

### Nouvelles fonctions dans la version 10.2

La liste ci-dessous répertorie les nouvelles fonctions depuis la dernière édition.

## Nouvelles fonctions de bases de données

Les fonctions propres à Vectorwise s'affichent désormais dans l'éditeur d'expression dans le dossier Fonctions de bases de données.

---

## Fonctions modifiées dans la version 10.2

La liste ci-dessous répertorie les modifications apportées aux fonctions depuis la dernière édition. Des liens vers des rubriques connexes sont inclus.

### Modification de la langue de l'interface utilisateur du produit

Pour Framework Manager, vous pouvez spécifier la langue de l'interface utilisateur du produit.

Par défaut, la langue du produit Framework Manager est la langue du système. Vous pouvez cependant modifier cette langue en ajoutant une entrée de préférence dans le fichier bmt.ini. Par exemple,

```
>Section name = BMTUI  
>Preference Name="ForceProductLocale">en-ca
```

Remplacez la chaîne "en-ca" par tout code de langue valide. Ce code indique la langue et/ou les paramètres régionaux à utiliser pour certaines sections de l'interface du produit, telles que les commandes de menu.

### Spécification du mode de requête d'un projet

Lorsque vous créez un projet dans IBM Cognos Framework Manager, le nouveau projet utilise à présent le mode de requête dynamique par défaut.

### Modèles durables pris en charge par les sources de données DMR

Les traitements de modèles durables sont à présent pris en charge pour la modélisation dimensionnelle des sources de données relationnelles (DMR).

---

## Chapitre 2. Initiation à l'utilisation de Framework Manager

IBM Cognos Framework Manager est un outil de modélisation des données qui permet de générer des requêtes pour IBM Cognos Software. Un modèle est un ensemble de métadonnées comprenant des informations physiques et professionnelles pour une ou plusieurs sources de données. IBM Cognos Software permet la gestion des performances sur des données relationnelles normalisées ou dénormalisées, ainsi que sur diverses sources de données OLAP. Lorsque vous ajoutez des fonctions de sécurité et des fonctions multilingues, un modèle peut répondre aux besoins de génération de rapports, d'interrogation ponctuelle et d'analyse de nombreux groupes d'utilisateurs à travers le monde.

Avant de faire quoi que ce soit dans IBM Cognos Framework Manager, vous devez parfaitement comprendre le problème de génération de rapports que vous tentez de résoudre.

Pour commencer, procédez comme suit :

### Procédure

1. Analysez le problème de génération de rapports.
2. Renseignez-vous sur les objets que vous utiliserez.
3. Créez et ouvrez un projet.
4. Explorez les sous-fenêtres de Framework Manager.
5. Explorez les exemples de modèle inclus avec Framework Manager.

---

## Analyse du problème

Avant de commencer, vous devez comprendre le problème de génération de rapports que vous tentez de résoudre et savoir de quelles données vous disposez pour cela.

Si vous ne parvenez pas à répondre aux questions suivantes, discutez avec vos utilisateurs de leurs besoins de génération de rapports :

### Procédure

1. Les utilisateurs et vous-même êtes-vous d'accord sur les exigences de génération de rapports ?  
Les problèmes à résoudre peuvent inclure le multilinguisme, les performances, la sécurité et la façon d'organiser et de combiner des éléments de requête et des filtres.
2. La source de données contient-elle les données et métadonnées dont vous avez besoin ?  
Sans métadonnées telles que des clés primaires, des index et des clés externes, les rapports peuvent être trop longs à exécuter ou produire des résultats incorrects. Si la source de données ne contient pas les données et métadonnées dont vous avez besoin, va-t-elle être modifiée, ou allez-vous trouver une solution de contournement ?
3. Les mêmes données existent-elles dans plusieurs sources ?  
Si tel est le cas, sélectionnez la source de données la mieux adaptée à vos besoins de génération de rapports. Si un entrepôt de données est disponible, il

constitue généralement un choix plus judicieux qu'une base de données opérationnelle. Un entrepôt de données basé sur un schéma en étoile est idéal. Si vous n'en disposez pas et que vous pensez que votre application de génération de rapports sera fortement sollicitée, cela vaut peut-être la peine d'en créer un.

4. Quelles sont les tables de source de données qui tiennent lieu de tables de faits, de dimensions, et à la fois de table de faits et de dimensions ?
5. Quels sont les clés et attributs de chaque dimension ?
6. Quelles sont les relations requises ?
7. Existe-t-il plusieurs chemins de relation entre des tables ?  
Si tel est le cas, que représente chaque chemin ? Vous devez définir le chemin préféré pour chacune des tables.

## Résultats

Vous devez vérifier que vous n'utilisez pas de noms réservés par IBM Cognos pour les noms des sources de données, des tables et des colonnes. Si vous devez utiliser un mot réservé, placez-le entre guillemets dans la spécification SQL. Par exemple, `select Orderdate, "Timezone"`. Pour plus d'informations, voir Annexe G, «Mots réservés», à la page 669.

---

## Création d'applications IBM Cognos Business Intelligence

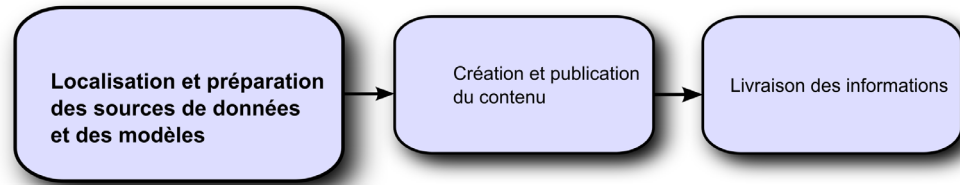
Vous utilisez les composants IBM Cognos Business Intelligence pour créer des applications de génération de rapport et d'analyse.

La durée de vie d'une application IBM Cognos Business Intelligence peut s'exprimer en mois ou même en années. Pendant cette période, les données peuvent être modifiées et de nouveaux critères peuvent apparaître. Comme les données sous-jacentes changent, les auteurs doivent modifier le contenu existant et en développer un nouveau. Les administrateurs doivent aussi mettre à jour les modèles et les sources de données au fil du temps. Pour en savoir davantage sur l'utilisation des sources de données, reportez-vous au *Guide d'administration et de sécurité d'IBM Cognos Business Intelligence* et au *Guide d'utilisation d'IBM Cognos Framework Manager*.

### Avant de commencer

Dans une application opérationnelle, l'infrastructure technique et de sécurité et le portail sont en place, ainsi que les processus de gestion des modifications, de contrôle des données, etc. Pour en savoir davantage sur le flux de travaux associé à la création de contenu IBM Cognos BI, reportez-vous au manuel *IBM Cognos Business Intelligence Architecture and Deployment Guide*. Pour en savoir davantage, reportez-vous à la boîte à outils IBM Cognos Solutions Implementation Methodology, qui comprend une documentation relative à l'implémentation et des documents d'aide. Des informations sur la boîte à outils sont disponibles sur le site IBM Support Portal (<http://www.ibm.com/software/analytics/support/finding-answers>).

Lorsque vous utilisez IBM Cognos BI pour créer des applications dans tous vos composants IBM Cognos BI, vous localisez et préparez des sources de données et des modèles, générez et publiez le contenu et distribuez les informations. Le graphique suivant fournit une vue d'ensemble du flux de travaux.



## Procédure

1. Localisez et préparez les sources de données et les modèles.

IBM Cognos BI permet de créer des rapports à partir d'une grande variété de sources de données, à la fois relationnelles et dimensionnelles. Les connexions à la base de données sont créées dans l'interface d'administration Web et servent à la modélisation, la création et l'exécution de l'application.

Pour pouvoir utiliser les données pour la création et l'affichage, les studios de Business Intelligence ont besoin d'un sous-ensemble d'un modèle des métadonnées (appelé pack). Les métadonnées peuvent nécessiter un processus de modélisation étendu dans Framework Manager.

2. Créez et publiez le contenu.

Des rapports, des scorecards, des analyses, des espaces de travail, etc. sont créés dans les studios d'informations décisionnelles d'IBM Cognos BI. Le studio à utiliser dépend du contenu, de la durée de vie et des utilisateurs du rapport, ainsi que du type de modélisation appliqué aux données (dimensionnelle ou relationnelle). Par exemple, l'analyse et la génération de rapports libre-service s'effectuent à l'aide d'IBM Cognos Workspace Advanced, d'IBM Cognos Query Studio et d'IBM Cognos Analysis Studio, tandis que les rapports planifiés sont créés dans IBM Cognos Report Studio. Les rapports et scorecards Report Studio sont généralement destinés à un public plus large ; ils sont publiés dans IBM Cognos Connection ou un autre portail, où ils sont programmés en vue d'être diffusés en rafale, distribués, etc. Report Studio peut également servir à la préparation de modèles pour la génération de rapports libre-service.

3. Diffusez et affichez les informations

Vous pouvez transmettre le contenu du portail IBM Cognos ou d'autres portails pris en charge et afficher les informations enregistrées dans ces portails ou fournies d'une autre façon. Vous pouvez également exécuter des rapports, des analyses, des scorecards, etc. à partir du studio d'informations décisionnelles avec lequel ils ont été créés.

Pour plus d'informations sur l'optimisation et les performances, voir le *Guide d'administration et de sécurité IBM Cognos Business Intelligence* et le site IBM Support Portal (<http://www.ibm.com/software/analytics/support/finding-answers>).

---

## Objets que vous utiliserez

Lorsque vous travaillez dans IBM Cognos Framework Manager, vous travaillez avec un certain nombre d'objets qui sont contenus dans un projet.

### Projets

Un projet contient un modèle, des espaces-noms, des packs, des sources de données, ainsi que des informations connexes pour la gestion et le partage des informations sur le modèle. Un projet unique peut couvrir plusieurs sources de données ou tables.

Un projet IBM Cognos Framework Manager s'affiche sous la forme d'un dossier qui contient un fichier de projet (.cpf) et les fichiers .xml spécifiques qui définissent le projet. Les fichiers contenus dans un dossier de projet sont uniques pour chaque projet. Le projet et ses fichiers associés sont contenus dans un dossier de projet.

En règle générale, vous ne devez pas ajouter de fichiers secondaires au dossier de projet car ils peuvent être affectés par des actions telles que les commandes Déplacer, Renommer et Supprimer du menu **Gérer les projets**. Si vous décidez d'ajouter des fichiers secondaires aux dossiers de projet, ces fichiers sont ajoutés avec des chemins absolus. S'ils sont déplacés à partir de leur emplacement d'origine, ils doivent être reciblés.

Voici le contenu d'un dossier de projet.

**<nom du projet>.cpf**

Fichier de projet Framework Manager, qui fait référence aux fichiers .xsd et .xml qui définissent un projet.

**archive-log.xml**

Ce fichier contient la partie du fichier journal principal qui a été archivé.

**customdata.xml**

Ce fichier contient les informations de présentation du diagramme.

Si ce fichier est supprimé, les informations de présentation sont perdues. Une présentation automatique est appliquée.

**IDLog.xml**

Ce fichier effectue un suivi d'objets pour les modèles qui utilisent le branchement et la fusion.

**log.xml**

Liste de toutes les modifications apportées au modèle.

**mda\_metadata.xml**

Fichier Model Design Accelerator, qui contient les métadonnées importées à partir de sources de données.

**mda\_engine\_project.xml**

Fichier Model Design Accelerator, qui contient la définition du schéma en étoile.

**model.xml**

Données de modèle réelles créées par les utilisateurs de Framework Manager.

**preferences.xml**

Préférences définies pour les projets Framework Manager.

**session-log.xml**

Liste des transactions non enregistrées dans le modèle. Lorsque le projet est enregistré, cette liste est supprimée. Pour afficher le contenu de ce fichier, utilisez la vue de l'historique des transactions.

Lorsque Framework Manager est démarré, le fichier session-log.xml existant est renommé session-log-backup.xml.

**session-log-backup.xml**



Fichier `session-log.xml` issu de la session précédente. A l'aide de ce fichier, un modélisateur peut exécuter un script afin de restaurer les transactions de modèle non enregistrées si une interruption inattendue se produit dans la session en cours.

Ce fichier est supprimé à chaque démarrage de Framework Manager. Prenez soin de faire une copie de ce fichier avant de quitter la session Framework Manager en cours si vous souhaitez conserver une copie.

#### **repository.xml**

Historique de version consigné pour chaque projet ou segment ayant été ajouté au référentiel ; ce fichier existe uniquement si vous avez ajouté des projets à un référentiel.

#### **upgradeReport.htm**

Contenu du message récapitulatif de mise à niveau qui s'affiche après la mise à niveau.

## **Modèles**

Un modèle est l'ensemble des dimensions, sujets de requête et autres objets connexes requis pour une ou plusieurs applications de génération de rapports associées.

Le modèle Framework Manager est une couche de métadonnées qui ajoute de la valeur à une source de données de différentes façons. Plus important encore, il fournit une vue métier des informations contenues dans les données source pour simplifier la création des rapports, des analyses et des requêtes. La vue métier peut :

- organiser les éléments dans des dossiers représentant des domaines d'activité pour la génération de rapports ;
- formater des éléments à l'aide de formats numérique, de devise, de date, d'heure et autres ;
- présenter des noms de dossier et d'élément, des descriptions, des astuces et des données multilingues de sorte que les utilisateurs puissent travailler dans la langue de leur choix ;
- automatiser la génération des requêtes SQL envoyées à la source de données relationnelles ;
- spécifier la fonction d'invite par défaut.

En l'occurrence, IBM Cognos peut afficher une invite pour l'utilisateur à l'aide d'un nom descriptif tout en effectuant un filtrage sur un code ou une valeur de clé pour améliorer les performances de requête.

Plus spécifiquement, vous pouvez modifier le modèle Framework Manager pour faire en sorte que les requêtes envoyées à la source de données soient efficaces, bien formées et sécurisées. Vous pouvez spécifier les règles qui régissent la génération des requêtes, limiter l'accès des utilisateurs à certaines lignes ou colonnes de données et modéliser des relations de données pour masquer la complexité des données de vos utilisateurs.

## **Espaces-noms**

Un espace-noms identifie de manière unique des éléments de requête, des dimensions, des sujets de requête et d'autres objets. Vous pouvez importer différentes bases de données dans des espaces-noms distincts afin d'éviter les doublons.

## Packs

Un pack est un sous-ensemble des dimensions, sujets de requête et autres objets définis dans le projet. Un pack représente ce qui est réellement publié sur le serveur IBM Cognos BI et il permet de créer des rapports, des analyses et des requêtes ad hoc.

## Dimensions

Une dimension est un groupement élargi de données concernant un aspect essentiel d'une entreprise, tel que des produits, des dates ou des marchés.

Les types de dimension que vous pouvez gérer dans IBM Cognos Framework Manager sont des dimensions ordinaires et des dimensions de mesure. Dans SAP BW, les dimensions de mesure sont appelés chiffres clés.

## Sujets de requête

Un sujet de requête est un ensemble d'éléments de requête liés par une relation intrinsèque.

Dans la plupart des cas, les sujets de requête se comportent comme des tables. Les sujets de requête produisent le même ensemble de lignes, quelles que soient les colonnes qui font l'objet de la requête.

Il existe différents types de sujet de requête.

### Source de données

Les sujets de requête de source de données font directement référence à des données d'une source de données unique. IBM Cognos Framework Manager crée automatiquement un sujet de requête de source de données relationnelles pour chacune des tables et vues que vous importez dans votre modèle.

### Modèle

Les sujets de requête de modèle ne sont pas générés directement à partir d'une source de données, mais ils sont basés sur des éléments de requête contenus dans d'autres sujets de requête ou dimensions, y compris d'autres sujets de requête de modèle. En utilisant des sujets de requête de modèle, vous pouvez créer une vue orientée métier plus abstraite d'une source de données.

### Procédure stockée

Les sujets de requête de procédure stockée sont générés lorsque vous importez une procédure à partir d'une source de données relationnelles. IBM Cognos Framework Manager ne prend en charge que les procédures stockées définies par l'utilisateur. Les procédures stockées définies par le système ne sont pas prises en charge.

## Éléments de requête

Un élément de requête est le plus petit élément du modèle qui peut être placé dans un rapport. Il représente une caractéristique unique d'un élément, telle que la date à laquelle un produit a été mis sur le marché.

Les éléments de requête sont contenus dans des sujets de requête ou des dimensions. Par exemple, un sujet de requête qui référence à une table entière contient des éléments de requête représentant chaque colonne de la table.

Pour les utilisateurs, les éléments de requête sont les objets les plus importants lors de la création de rapports. Ils utilisent les propriétés des éléments de requête pour générer leurs rapports.

---

## Création de projets

Dans IBM Cognos Framework Manager, vous travaillez dans le contexte d'un projet. Le projet contient des objets que vous organisez pour les utilisateurs en fonction du modèle de gestion et des règles métier de votre organisation. Vous affichez ces objets dans la page du projet.

### Avant de commencer

Avant de pouvoir importer des métadonnées, vous devez créer un projet.

Pour plus d'informations sur la création d'un segment de projet, voir «Création de segments», à la page 286.

Pour plus d'informations sur la création d'un projet à l'aide de Model Design Accelerator, voir «Création d'un projet», à la page 32.

### Procédure

1. Dans la page **Bienvenue**, cliquez sur **Créer un nouveau projet**.

**Astuce** : Si vous êtes dans Framework Manager, cliquez sur **Fichier, Nouveau**.

2. Dans la page **Nouveau projet**, spécifiez un nom et un emplacement pour le projet et cliquez sur **OK**.

Par défaut, un nouveau projet utilise un mode de requête dynamique. Pour utiliser le mode de requête compatible, cochez la case **Utiliser le mode de requête dynamique**. Pour plus d'informations, reportez-vous au *Guide sur les requêtes dynamiques* d'IBM Cognos.

3. Dans la page **Sélectionner la langue**, cliquez sur la langue de conception du projet.

Vous ne pouvez pas modifier la langue après avoir cliqué sur **OK**, mais vous pouvez ajouter d'autres langues de projet. Pour plus d'informations, voir «Ajout d'une langue à un projet», à la page 129.

Pour plus d'informations sur la sélection de la langue de conception appropriée pour les modèles durables, voir «Modèles durables», à la page 187.

**Remarque** : Si un serveur SAP BW ne prend pas en charge la langue sélectionnée, il utilise le mappage des paramètres régionaux de contenu d'IBM Cognos Configuration. Si un mappage n'est pas défini, Framework Manager utilise la langue par défaut du serveur SAP BW.

4. Cliquez sur **OK** pour sélectionner la langue de conception.

L'**assistant de métadonnées** s'affiche.

5. Choisissez si vous souhaitez importer les métadonnées maintenant ou ultérieurement.

- Pour les importer maintenant, sélectionnez la source d'importation et cliquez sur **Suivant**.
- Pour reporter l'importation des métadonnées, cliquez sur **Annuler**.

6. Si vous avez choisi d'importer maintenant les métadonnées, suivez les instructions de l'**assistant de métadonnées** :
  - Sélectionnez une connexion de source de données et cliquez sur **Suivant**.  
Si la connexion de source de données souhaitée n'est pas répertoriée, vous devez d'abord la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.
  - Cochez les cases correspondant aux objets que vous souhaitez importer.
  - Spécifiez la manière dont l'importation doit traiter les noms d'objet en double. Indiquez si vous souhaitez importer et créer un nom unique. Si vous choisissez de créer un nom unique, l'objet importé s'affiche avec un numéro. Par exemple, vous verrez SujetRequête et SujetRequête1 dans votre projet.
  - Si vous souhaitez importer des objets système, cochez la case **Afficher les objets système**, puis sélectionnez les objets système à importer.
  - Spécifiez les critères à utiliser pour créer des relations et cliquez sur **Importer**.  
Pour plus d'informations, voir «Relations», à la page 68.Une liste des objets qui n'ont pas pu être importés et le nombre d'objets importés s'affichent.
7. Cliquez sur **Terminer**.  
Enregistrez le fichier projet (.cpf) et tous les fichiers associés dans un seul dossier. Si vous enregistrez un projet sous un autre nom ou un autre format, veillez à enregistrer le projet dans un dossier distinct.

---

## Ouverture de projets

Vous devez ouvrir un projet pour pouvoir importer des métadonnées ou modifier des métadonnées existantes.

Si le projet a été créé à l'aide d'un schéma de modèle antérieur à la version actuellement prise en charge, vous pouvez être invité à mettre à niveau le modèle.

Si votre modèle est restitué dans un référentiel, vous ne pouvez pas mettre à niveau. Extrayez manuellement le modèle depuis le système de contrôle source, puis ouvrez-le dans la nouvelle version d'IBM Cognos Framework Manager.

Si vous mettez à niveau un modèle segmenté, vous devez ouvrir et mettre à niveau chaque segment séparément. Après la mise à niveau de chaque segment, vous pouvez mettre à niveau le projet de niveau supérieur, ou principal.

### Procédure

1. Dans la page **Bienvenue**, cliquez sur **Ouvrir un projet**.  
**Astuce** : Si vous êtes dans Framework Manager, cliquez sur **Fichier, Ouvrir**.
2. Recherchez le dossier du projet et cliquez sur le fichier .cpf.
3. Cliquez sur **OK**.

---

## Page de projet
















Une fois que vous avez créé ou ouvert un projet, la page de projet s'affiche. C'est dans cette page que vous concevez, insérez dans un pack et publiez les métadonnées du projet. Cette page contient plusieurs sous-fenêtres et vues que vous pouvez utiliser pour afficher et modifier les objets d'un projet.




















## Afficheur de projet








L'afficheur de projet fait apparaître les objets d'un projet dans une vue hiérarchique. Vous pouvez utiliser l'afficheur de projet pour visualiser, modifier et créer des objets.

Les relations sont affichées dans les onglets **Diagramme** et **Explorateur**. Les fonctions importées s'affichent dans l'onglet **Explorateur**.

L'**afficheur de projet** utilise les icônes suivantes pour représenter des objets et des états. Un projet peut utiliser tout ou partie des icônes.

Icône	Objet
	Projet
	Espace-noms racine ou tout autre espace-noms du projet
	Dossier de source de données
	Source de données
	Dossier de table de mappage des paramètres
	Table de mappage des paramètres
	Pack
	Dossier de packs
	Pack publié
	Calcul
	Calcul dont la propriété <b>Usage</b> est définie sur <b>Attribut</b>
	Calcul dont la propriété <b>Usage</b> est définie sur <b>Identificateur</b>
	Calcul intégré
	Calcul dont la propriété <b>Usage</b> est définie sur <b>Fait</b>
	Dimension de modèle basée sur des objets de modèle existants

Icône	Objet
	Dimension importée à partir d'une source de données
	Dimension non valide. Cette dimension est importée à partir d'une source de données.
	Filtre
	Hiérarchie
	Niveau dans une hiérarchie
	Mesure
	Mesure semi-additive
	Dimension de de mesure basée sur des objets de modèle existants.
	Dimension de mesure importée à partir d'une source de données.
	Dimension de mesure non valide. Cette dimension est importée à partir d'une source de données.
	Elément de requête
	Elément de requête dont la propriété <b>Usage</b> est définie sur <b>Identificateur</b>
	Elément de requête dont la propriété <b>Usage</b> est définie sur <b>Fait</b>
	Elément de requête situé sous un sujet de requête de raccourci
	Sujet de requête basé sur des objets de modèle existants
	Sujet de requête importé à partir d'une source de données
	Sujet de requête non valide. Ce sujet de requête est importé à partir d'une source de données.
	Sujet de requête basé sur des données multidimensionnelles
	Sujet de requête dont la propriété <b>Usage</b> est définie sur <b>Fait</b> . Ce sujet de requête est basé sur des objets de modèle existants.

Icône	Objet
	Sujet de requête dont la propriété <b>Usage</b> est définie sur <b>Fait</b> . Ce sujet de requête est importé à partir d'une source de données.
	Sujet de requête dont la propriété <b>Usage</b> est définie sur <b>Fait</b> . Ce sujet de requête est importé à partir d'une source de données et n'est pas valide.
	Relation
	Segment lié ou projet qui a été mis à jour. Cette icône s'affiche par dessus les autres icônes.
	Objet lié. Cette icône s'affiche par dessus les autres icônes.
	Raccourci. Cette icône s'affiche par dessus les autres icônes.
	Objet non valide. Cette icône s'affiche par dessus les autres icônes.

## Modification d'options pour des projets

Vous pouvez modifier une ou plusieurs options pour tester des projets et enregistrer automatiquement les modifications.

Les options de test s'appliquent globalement à toutes les séries de tests pour ce projet. Pour plus d'informations sur le d'objets sélectionnés, voir Chapitre 5, «Modélisation des métadonnées relationnelles», à la page 67 ou Chapitre 6, «Métadonnées SAP BW», à la page 197.

L'enregistrement automatique des projets peut aider à éviter la perte de données si une interruption imprévue, comme une coupure de courant, a lieu. Si vous choisissez d'enregistrer automatiquement les projets, vous pouvez sélectionner l'intervalle de temps entre les opérations de sauvegarde.

Si vous optez pour la fonction d'enregistrement automatique, IBM Cognos Framework Manager ne peut pas enregistrer les modifications lorsqu'une boîte de dialogue est ouverte si cette boîte de dialogue verrouille d'autres zones de Framework Manager. Par exemple, lorsque vous utilisez les boîtes de dialogue présentées par l'assistant **Créer un dossier**, vous ne pouvez pas accéder aux commandes de la barre de menus. Une fois que vous avez fermé la boîte de dialogue, Framework Manager enregistre automatiquement les modifications une fois l'intervalle d'enregistrement automatique écoulé.

### Procédure

1. Cliquez sur **Projet, Options**.
2. Dans l'onglet **Options de test**, sélectionnez les options voulues.

Objectif	Action	Persistance
<p>Limiter le nombre de lignes extraites.</p>	<p>Cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes.</p> <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>	<p>Ce paramètre s'applique à toutes les dimensions, tous les sujets de requête et tous les ensembles de requêtes du modèle.</p> <p>Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel projet.</p>
<p>Indiquer le niveau de détail</p>	<p>Faites glisser le curseur de <b>Niveau de détail affiché dans les informations de requête</b> vers l'emplacement qui représente la quantité d'informations dont vous avez besoin.</p>	<p>Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce projet.</p>
<p>Appliquer les filtres de mode de conception appropriés</p>	<p>Cochez la case <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b>.</p> <p>Ce paramètre s'applique à tous les filtres pertinents dont l'utilisation est définie sur le mode de conception dans une autre dimension, un autre sujet de requête ou un autre ensemble de requêtes.</p>	<p>Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel projet.</p>

3. Si vous souhaitez que les projets soient enregistrés automatiquement, dans l'onglet **Enregistrement automatique**, sélectionnez **Activé** et entrez un nombre représentant la fréquence en minutes voulue pour l'enregistrement automatique des projets.
4. Cliquez sur **OK**.

## Réorganisation d'objets

Par défaut, les objets d'espace-noms racine répertoriés dans l'**afficheur de projet** s'affichent dans l'ordre où ils ont été ajoutés. Vous pouvez changer l'ordre des objets triés selon leurs noms. Les objets peuvent être réorganisés en ordre croissant ou décroissant.

Lorsque vous sélectionnez des objets à réorganiser, toutes les sélections doivent être au même niveau dans l'arborescence de métadonnées. Cependant, vous pouvez réorganiser uniquement certains objets sur un niveau. L'emplacement des objets réorganisés par rapport aux objets qui n'ont pas été sélectionnés est basé sur la position du premier objet de la liste réordonnée. Par exemple, lors d'une réorganisation en ordre croissant, l'objet dont le nom commence par la lettre la plus proche "A" reste dans sa position actuelle dans l'**afficheur de projet**, suivi par les autres objets sélectionnés.



Vous pouvez inclure les enfants lorsque vous réorganisez les objets sélectionnés. Par exemple, supposons que deux sujets de requête, Détaillants Ouest et Détaillants Est, sont affichés sur un niveau. Chaque sujet de requête comporte des enfants qui s'affichent un niveau au-dessous de leurs parents. Si vous réorganisez Détaillants Ouest et Détaillants Est et que vous choisissez d'inclure les enfants, l'**afficheur de projet** répertorie Détaillants Ouest et Détaillants Est sur un niveau. Les enfants de chaque sujet de requête sont regroupés sous leurs parents et sont répertoriés en ordre croissant.

Vous pouvez aussi choisir d'inclure les descendants des objets enfant. Le temps nécessaire pour réorganiser des objets peut être plus long selon le type et le nombre d'objets sélectionnés.

Si les sélections comprennent des objets ou des niveaux en lecture seule dans des hiérarchies, IBM Cognos Framework Manager affiche un message indiquant que ces éléments ne peuvent pas être réorganisés.

### Procédure

1. Dans l'**afficheur de projet**, sélectionnez les objets d'espace-noms racine à réorganiser.  
Vérifiez que les objets sélectionnés se trouvent au même niveau.  
Vous pouvez également sélectionner des objets dans l'onglet **Explorateur** ou l'onglet **Diagramme**.
2. Cliquez sur **Outils, Réorganiser**.
3. Cliquez pour indiquer si vous souhaitez réorganiser les objets par nom en ordre croissant ou décroissant.
4. Indiquez si vous souhaitez réorganiser les objets sélectionnés uniquement, ou les objets sélectionnés et leurs enfants.  
Si vous choisissez de réorganiser les enfants des objets sélectionnés, vous pouvez également inclure tous les descendants des objets enfant.
5. Cliquez sur **OK**.

## Onglet Explorateur

L'onglet Explorateur affiche le contenu d'un projet, semblable à n'importe quel système de fichiers. Organisez les objets par nom, par classe ou par description.


Si vous disposez d'un projet comportant grand nombre d'objets, il sera peut-être plus facile de les retrouver dans l'onglet **Explorateur**.

Vous pouvez utiliser l'onglet **Explorateur** pour afficher, créer et modifier des objets et des relations. Vous pouvez également créer des dossiers et des espaces-noms pour regrouper des objets.

## Onglet Diagramme

L'onglet **Diagramme** permet d'afficher les relations entre les objets d'un projet. Les relations entre les objets sont affichées sous forme de lignes avec une notation de cardinalité. Vous pouvez développer des objets et des espaces-noms pour afficher la hiérarchie des objets et les relations entre les objets.

**Astuce** : Vous pouvez également contrôler la zone de modèle qui est visible dans le diagramme. Tout en cliquant sur le bouton de **présentation** en bas à droite,

faites glisser le pointeur sur le diagramme .

Dans l'onglet **Diagramme**, vous pouvez effectuer les opérations suivantes :

- Afficher, créer et modifier des objets et des relations.
- Créer des dossiers et des espaces-noms pour regrouper des objets.
- Modifier les paramètres des diagrammes.
- Modifier la présentation des objets et afficher ces derniers en étoile ou dans une présentation standard en cliquant sur **Diagramme, Présentation automatique**.
- Mettre le focus sur un objet en cliquant sur **Diagramme, Définir le point focal**.
- Rechercher un objet en cliquant avec le bouton droit de la souris sur l'objet dans l'**afficheur de projet** et en cliquant sur **Rechercher dans le diagramme d'objets**.
- Effectuer un zoom avant ou arrière en cliquant sur **Diagramme, Zoom**.
- Développer ou réduire tous les objets à partir du menu **Diagramme**.
- Enregistrer le diagramme pour impression en cliquant sur **Diagramme, Capture d'écran** et en indiquant le nom de l'image.
- Lancer **Context Explorer** en cliquant avec le bouton droit de la souris sur un objet dans l'onglet **Diagramme** et en cliquant sur **Lancer Context Explorer**.
- Imprimer le diagramme dans **Context Explorer** en cliquant avec le bouton droit de la souris sur l'arrière-plan **Context Explorer** et en cliquant sur **Imprimer**. Un bouton d'impression est également disponible dans la barre d'outils de **Context Explorer**. Ce menu contextuel inclut également des commandes pour prévisualiser le diagramme avec **Aperçu avant impression** et modifier les options de mise en page avec **Mise en page**.

## Modification des paramètres de diagrammes

Vous pouvez modifier un ou plusieurs paramètres pour des diagrammes. L'onglet **Paramètres de diagramme** définit les paramètres pour le diagramme principal. L'onglet **Context Explorer** définit la manière dont les diagrammes de contexte sont alimentés et affichés.

### Procédure

1. Cliquez sur **Diagramme, Paramètres de diagramme**.
2. Définissez le niveau de détail.  
Vous pouvez sélectionner des éléments de requête, des relations, des relations de portée, une cardinalité et des descriptions.
3. Sélectionnez le type de notation.  
Vous pouvez utiliser la notation Merise ou Crowsfeet. Par défaut, IBM Cognos Framework Manager utilise la notation Merise. La notation Merise marque chaque extrémité de la relation avec la cardinalité minimale et maximale de cette extrémité. Vous pouvez également utiliser la notation Crowsfeet qui fournit une représentation illustrée de la relation.
4. Pour aligner plus facilement les objets, activez l'option **Options d'instantané**.
5. Pour disposer d'un réseau de lignes espacées régulièrement en arrière-plan, sélectionnez la case à cocher **Afficher la grille**.
6. Sélectionnez la police et la couleur du texte.
7. Indiquez si vous souhaitez que ces paramètres deviennent les paramètres par défaut pour tous les nouveaux projets.
8. Cliquez sur **OK**.

## Onglet Grille des dimensions

Vous pouvez utiliser l'onglet **Grille des dimensions** pour afficher, créer et modifier des hiérarchies et des niveaux pour la dimension que vous avez sélectionnée dans l'**afficheur de projet**. Vous pouvez également afficher et modifier les relations de portée.

Les onglets **Mesures** et **Attributs** s'affichent lorsque vous cliquez sur l'onglet **Grille des dimensions**. L'onglet **Mesures** permet d'afficher ou de modifier toutes les mesures et les relations de portée qui sont disponibles dans le modèle. L'onglet **Attributs** permet d'afficher ou de modifier le rôle de l'élément de requête sélectionné.

Vous pouvez également incorporer des calculs dans l'élément de requête.

**Astuce :** Pour afficher au mieux les métadonnées SAP BW, utilisez la présentation en étoile. Dans le menu **Diagramme**, cliquez sur **Présentation automatique, Etoile**.

## Sous-fenêtre Propriétés

La sous-fenêtre **Propriétés** affiche les propriétés des objets que vous avez sélectionnés en dernier dans l'onglet **Afficheur de projet**, **Explorateur**, **Diagramme**, **Grille des dimensions** ou **Dépendances**, ou dans la sous-fenêtre **Rechercher** ou **Récapitulatif**. Les propriétés d'objet sont définies lors de l'importation et certaines valeurs de propriété peut être modifiées pendant la modélisation. Vous pouvez utiliser la sous-fenêtre **Propriétés** pour ajouter, modifier ou supprimer les propriétés d'objets.

Vous pouvez modifier les propriétés de plusieurs objets en même temps. Si vous sélectionnez plusieurs objets, IBM Cognos Framework Manager n'affiche que les propriétés qui sont communes à tous les objets. Vous pouvez :

- Trier des valeurs de propriété en cliquant deux fois sur l'en-tête d'une propriété  
Une flèche s'affiche pour indiquer le sens dans lequel les valeurs sont triées. Vous pouvez basculer entre l'ordre croissant et l'ordre décroissant.
- Filtrer les valeurs de propriété en cliquant sur la flèche située à droite de l'en-tête d'une propriété  
Vous pouvez cliquer sur une valeur ou sur **Personnalisé** pour définir les critères pour les lignes que vous souhaitez afficher.
- Appliquer la valeur d'une propriété à plusieurs objets en cliquant sur la flèche en regard de de la propriété et faisant glisser la zone mise en évidence sur les propriétés auxquelles vous souhaitez appliquer cette valeur
- Redimensionner la largeur des lignes et des colonnes en cliquant avec le bouton droit de la souris sur le nom d'un objet dans la sous-fenêtre Propriétés

Si vous avez besoin de plus d'espace, vous pouvez redimensionner la sous-fenêtre **Propriétés** ou la déplacer pour qu'elle flotte dans la fenêtre. Par exemple, si vous disposez de plusieurs écrans, vous pouvez déplacer la sous-fenêtre **Propriétés** vers un autre écran.

### Remplacement de plusieurs valeurs de propriété

Vous pouvez remplacer plusieurs valeurs pour chaque propriété de chaîne de texte.

Lorsque vous remplacez les valeurs de plusieurs propriétés, IBM Cognos Framework Manager met automatiquement à jour les éléments qui dépendent des valeurs que vous avez remplacées. Si des dépendances ne peuvent pas être mises à

jour, par exemple, des noms de source de données et de table de mappage des paramètres, vous devez valider manuellement le modèle.

Vous pouvez faire correspondre la totalité de la valeur de propriété, ou seulement une partie. En mettant en correspondance des parties de la valeur de propriété, vous pouvez effectuer plusieurs remplacements sur une seule propriété de texte. Par exemple, si vous recherchez "nation" pour remplacer ce terme par "région", la valeur de la propriété "Cette nation et cette nation" est remplacée par "Cette région et cette région". Cependant, "national" va être remplacé par "régional".

**Astuce :** Avant de remplacer des valeurs de propriété, élargissez la sous-fenêtre **Propriétés** et la colonne de la propriété dont vous souhaitez remplacer la valeur.

### Procédure

1. Dans la fenêtre **Afficheur de projet**, sélectionnez plusieurs objets.

**Conseil :** Filtrez d'abord les propriétés pour réduire l'ensemble de propriétés à rechercher avant d'effectuer le remplacement. Dans l'onglet **Propriétés**, cliquez avec le bouton droit de la souris sur l'en-tête de la colonne que vous voulez filtrer et cliquez sur **Définir le filtrage automatique**.

2. Effectuez l'une des opérations suivantes :
  - Si votre modèle contient plusieurs langues, cliquez sur l'onglet **Langue**.
  - Si votre modèle contient une seule langue, cliquez sur l'onglet **Propriétés**.
3. Cliquez avec le bouton droit sur l'en-tête de colonne correspondant à la propriété dont vous voulez remplacer les valeurs, puis cliquez sur **Remplacement en bloc**.
4. Dans les zones **Rechercher** et **Remplacer par**, entrez les chaînes de texte de recherche et de remplacement.

**Conseil :** Pour remplacer des propriétés vides, laissez la zone **Rechercher** vide.

5. Pour effectuer une recherche sensible à la casse, cochez la case **Respecter la casse**.
6. Pour rechercher le texte de propriété complet, cochez la case **Correspondre à tout le contenu des cellules**.

**Conseil :** Pour remplacer toutes les zones de propriété, quel que soit le texte qu'elles contiennent, tapez un astérisque dans la zone **Rechercher** et cochez la case **Correspondre à tout le contenu des cellules**.

7. Effectuez une ou plusieurs des opérations suivantes :
  - Cliquez sur **Remplacer** pour remplacer l'objet actuellement sélectionné et rechercher l'occurrence suivante.
  - Cliquez sur **Suivant** pour sélectionner l'objet correspondant suivant.
  - Cliquez sur **Remplacer tout** pour remplacer toutes les propriétés correspondantes à partir de la première propriété correspondante dans la liste.
  - Cliquez sur **Fermer** si vous avez terminé le remplacement.

Au fur et à mesure que des modifications sont effectuées sur une propriété, les résultats s'affichent dans l'onglet.

Lorsque le remplacement en bloc a atteint le dernier objet de la liste, un message s'affiche indiquant le nombre de remplacements.

## Sous-fenêtre Outils

La sous-fenêtre **Outils** contient les onglets **Récapitulatif**, **Rechercher** et **Dépendances**.

Vous pouvez changer la sous-fenêtre **Outils** en une fenêtre distincte. Pour ce faire, cliquez sur la barre de titre de la sous-fenêtre et faites-la glisser dans n'importe quelle direction jusqu'à ce sa nomenclature devienne une ligne épaisse. Vous pouvez à présent faire glisser la fenêtre **Outils** vers n'importe quel emplacement sur votre bureau. Pour rétablir la fenêtre en une sous-fenêtre au sein de la fenêtre IBM Cognos Framework Manager, faites-la glisser vers l'une des bordures Framework Manager jusqu'à ce que sa nomenclature devienne une ligne fine.

### Onglet Récapitulatif

L'onglet **Récapitulatif** affiche la langue, des statistiques et les tâches disponibles pour l'objet sélectionné dans l'**afficheur de projet**.

La section **Projet** indique la langue de conception et la langue active. Vous pouvez changer de langue active.

La section **Statistiques** indique le nombre d'objets, par classe, se trouvant dans l'objet actuellement sélectionné. Si l'objet sélectionné contient un dossier, le contenu du dossier est inclus dans ce nombre. Les objets sélectionnés incluent les projets, les espaces-noms et les dossiers. L'objet sélectionné par défaut est le projet.

La section **Tâches** affiche les actions qui sont applicables à l'objet actuellement sélectionné, en fonction de sa classe. Si vous sélectionnez un dossier, les actions de dossier sont répertoriés. Si vous sélectionnez un objet dans ce dossier, la liste inclut les actions applicables à l'objet et au dossier.

Pour plus d'informations sur les classes d'objets, voir le fichier `c10_location\templates\bmt\CR1Model\BMTModelSpecification.xsd`.

### Utilisation de l'onglet Rechercher

Lorsque vous travaillez sur un projet de grande taille, il peut être difficile de localiser les objets dont vous avez besoin pour effectuer une tâche. Utilisez l'onglet **Rechercher** pour trouver rapidement des objets en appliquant différents critères de recherche, tels que l'emplacement, la classe, la condition ou la propriété.

### Avant de commencer


Si votre modèle contient plusieurs langues, veillez à spécifier la valeur **Langue active** dans la boîte de dialogue **Définir les langues** avant de rechercher des objets.

### Procédure

1. Cliquez sur **Outils**, **Rechercher**.

#### Astuces :

- Si la sous-fenêtre **Outils** n'est pas visible, cliquez sur **Vue**, **Outils**. Vous pouvez faire glisser la sous-fenêtre **Outils** vers le bas de la fenêtre Framework Manager et la redimensionner pour obtenir une meilleure vue des résultats de la recherche.
- Filtrez d'abord les propriétés pour réduire l'ensemble de propriétés de recherche. Dans l'onglet **Propriétés**, cliquez avec le bouton droit de la souris sur l'en-tête de la colonne que vous voulez filtrer et cliquez sur **Définir le filtrage automatique**.

2. Dans la zone **Chaîne de recherche**, saisissez la text à rechercher.  
Vous pouvez utiliser des chaînes en majuscules, en minuscules ou en casse mixte. La recherche n'est pas sensible à la casse. Les caractères génériques valides sont "\*" et "?".
3. Cliquez sur la **flèche double vers le bas**  pour afficher les zones correspondant aux critères de recherche.
4. Dans la liste **Condition**, sélectionnez une condition à appliquer à la chaîne de recherche.  
La zone **Condition** détermine la manière dont la valeur de **Chaîne de recherche** est mise en correspondance avec le texte dans le modèle. Elle contient une liste de conditions de recherche possibles. Si vous souhaitez effectuer une recherche à l'aide de caractères génériques, utilisez la condition **égal à**.  
Une expression régulière est une méthode complexe et puissante de mise en correspondance de chaînes de texte. Pour effectuer une recherche à l'aide d'une expression régulière, utilisez la condition **expression régulière**. Une recherche d'expression régulière est sensible à la casse. Par exemple, pour trouver tous les objets contenant le mot "Car", effectuez une recherche sur la chaîne "\Car", dans les guillemets.  
La valeur de la zone **Condition** est enregistrée de session en session.
5. Dans la liste **Rechercher dans**, sélectionnez la partie de la hiérarchie de modèle dans laquelle vous souhaitez effectuer la recherche.  
La valeur de la zone **Rechercher dans** est enregistrée de session en session.
6. Dans la liste **Classe**, sélectionnez la classe unique d'objets dans laquelle vous souhaitez effectuer la recherche.
7. Dans la liste **Propriété**, sélectionnez le type de propriété à rechercher.  
La propriété (**Toutes les propriétés**) recherche toutes les propriétés. La propriété **Nom d'objet** limite la recherche au nom de chaque objet. La propriété **Propriétés du texte** effectue la recherche dans l'ensemble des propriétés qui contiennent des chaînes de texte, tels que Description ou Infobulle, mais sans inclure le nom d'objet.  
La valeur de la zone **Propriété** est enregistrée de session en session.
8. Cliquez sur **Rechercher**.  
Les résultats sont répertoriés en bas de la page **Rechercher**. Les résultats de la recherche peuvent contenir plusieurs valeurs, par exemple, des propriétés de texte, pour un seul objet. Lorsque vous cliquez sur un objet comportant plusieurs valeurs, toutes les valeurs qui appartiennent à cet objet sont sélectionnées. Si votre modèle contient plusieurs langues, les résultats de la recherche incluent des propriétés pour chaque langue du modèle.  
Une fois que vous avez effectué une recherche, la case à cocher **Sous-ensemble** devient disponible. Si vous cochez la case **Sous-ensemble**, la recherche suivante s'effectue sur les résultats de recherche existants. La case à cocher **Sous-ensemble** est désélectionnée après chaque recherche. Vous pouvez effectuer des recherches de sous-ensemble successives en cochant la case **Sous-ensemble**.
9. Pour afficher un objet dans l'**afficheur de projet**, cliquez sur un objet dans l'onglet **Rechercher**.
10. Pour afficher un objet dans le diagramme, cliquez avec le bouton droit de la souris sur un objet dans l'onglet **Rechercher** et cliquez sur **Rechercher dans le diagramme d'objets**.

## Résultats

Le bouton **Remplacement en bloc** de l'onglet **Rechercher** vous permet d'effectuer une opération de recherche et de remplacement sur l'ensemble d'objets des résultats de la recherche. Lorsque vous cliquez sur le bouton **Remplacement en bloc**, l'ensemble de résultats s'affiche dans l'onglet **Propriétés** et la boîte de dialogue **Remplacement en bloc** s'affiche. Dans cette boîte de dialogue, vous pouvez sélectionner la propriété à appliquer à l'opération de recherche et de remplacement.

Vous pouvez également lancer un remplacement en bloc à partir de la sous-fenêtre **Propriétés**. Pour plus d'informations, voir «Remplacement de plusieurs valeurs de propriété», à la page 17.

## Onglet Dépendances

L'onglet **Dépendances** affiche les objets qui dépendent d'un objet sélectionné.

---

## Conventions de dénomination pour les objets d'un projet

Tous les objets d'un projet doivent avoir un identificateur unique. La référence peut être constituée d'une ou de plusieurs parties, selon le type d'objet. Les parties comportent un nom et un emplacement d'objet dans la hiérarchie du projet, tel que cela est indiqué dans la langue par défaut du projet.

**Remarque** : si vous souhaitez affecter le même nom à deux dimensions ou sujets de requête dans un projet, ils doivent se trouver dans des espaces-noms distincts.

### Identificateurs en une partie

Certains objets d'un projet sont dotés d'un identificateur en une partie. L'identificateur en une partie doit être unique dans l'ensemble du projet, même si l'espace-noms contient d'autres espaces-noms. Les objets suivants sont dotés d'un identificateur en une partie :

- Espaces-noms
- Fonctions
- Raccourcis vers des espaces-noms
- Raccourcis vers des dossiers

### Identificateur en deux parties

Certains objets d'un projet sont dotés d'un identificateur en deux parties composé du nom de l'espace-noms contenant et du nom de l'objet. Le nom d'objet doit être unique dans l'espace-noms contenant. Les objets suivants sont dotés d'un identificateur en deux parties :

- Dimensions ordinaires
- Dimensions de mesure
- Sujets de requête
- Raccourcis vers des sujets de requête

Par exemple, un espace-noms ventesVA contient un sujet de requête nommé Produit. Le sujet de requête Produit porte le nom suivant (les crochets et les points constituent la syntaxe utilisée par Framework Manager pour les identificateurs d'objet) :

[go\_sales].[Product]

## Identificateurs en trois parties

Certains objets d'un projet sont dotés d'un identificateur en trois parties basé sur l'identificateur du sujet de requête contenant. Chaque nom doit être unique dans le sujet de requête contenant. Les objets suivants sont dotés d'un identificateur en trois parties :

- Hiérarchies
- Mesures
- Éléments de requête

Par exemple, un espace-noms ventesVA contient un sujet de requête nommé Produit et un élément de requête appelé Code de produit. L'élément de requête Code de produit porte le nom suivant (les crochets et les points constituent la syntaxe utilisée par IBM Cognos Framework Manager pour les identificateurs d'objet) :

[go\_sales].[Product].[Product Code]

## Identificateurs en quatre parties

Les niveaux d'un projet sont dotés d'un identificateur en quatre parties composé du nom d'espace-noms, du nom de dimension, du nom de hiérarchie et du nom de niveau.

Par exemple, un espace-noms entrepôt\_données\_VA contient une dimension appelée Compte. Une hiérarchie dans la dimension Compte est un bilan, lequel contient un niveau appelé Compte. Le niveau Compte porte le nom suivant (les crochets et les points constituent la syntaxe utilisée par Framework Manager pour les identificateurs d'objet) :

[go\_data\_warehouse].[Account dimension].[Balance sheet].[Account]

## Identificateurs en cinq parties

Certains objets d'un projet sont dotés d'un identificateur en cinq parties composé du nom d'espace-noms, du nom de dimension, du nom de hiérarchie, du nom de niveau et du nom d'élément de requête. Les identificateurs en cinq parties sont également utilisés pour les légendes et les clés métier dans les noms uniques de membres. Pour plus d'informations, voir «Noms uniques de membres», à la page 113.

Par exemple, un espace-noms entrepôt\_données\_VA contient une dimension appelée Compte. Une hiérarchie dans la dimension Compte est un bilan, lequel contient un niveau appelé Compte. Le niveau Compte contient un élément de requête appelé Nom de compte. L'élément de requête Nom de compte porte le nom suivant (les crochets et les points constituent la syntaxe utilisée par Framework Manager pour les identificateurs d'objet) :

[go\_data\_warehouse].[Account dimension].[Balance sheet].[Account].[Account name]

---

## Exemples de modèle

Vous pouvez explorer plusieurs exemples de modèle fournis avec IBM Cognos Framework Manager. Les exemples de modèle ont été créés à l'aide des instructions de modélisation des métadonnées.



Le formatage par défaut est défini pour les éléments de requête présentés dans chaque exemple de modèle. Les noms et les descriptions ont été traduits dans différentes langues. A l'aide de la table de mappage des paramètres `Language_lookup`, chaque utilisateur peut visualiser les noms et les descriptions de dossier et d'élément dans la langue de leur choix.

Pour plus d'informations sur les exemples de rapport et de pack, voir le document *Report Studio User Guide*.

## Modèle d'entrepôt de données Vacances et Aventure

Ce modèle contient des informations sur la situation financière et les ressources humaines de la société fictive Vacances et aventure. Il accède à une source de données relationnelle dimensionnelle.

Cet exemple de modèle est situé dans `c10_location\webcontent\samples\models\great_outdoors_warehouse`.

L'exemple de modèle contient les vues suivantes :

- Vue de base de données

Contient les objets qui ont été importés à partir de la source de données. Ces objets sont regroupés dans des espaces-noms correspondant à chaque domaine d'activité et un espace-noms appelé GO pour les tables qui sont communes à tous les domaines d'activité.

En outre, les sujets de requête de modèle, les raccourcis et les filtres sont organisés espaces-noms. Les espaces-noms s'affichent sous la forme de dossiers pour les utilisateurs. Cette organisation des informations en catégories métier pertinentes permet aux utilisateurs de trouver les informations dont ils ont besoin.

Toutes les jointures et tous les déterminants sont définis dans cette vue.

Quelques sujets de requête de modèle ont été ajoutés pour les utilisateurs qui ont besoin de déterminants et de tables de correspondance.

- Vue métier

Contient des sujets de requête de modèle qui représentent les données en termes de besoins métier ou d'application.

Les calculs, y compris ceux pour le langage, sont définis dans cette vue.

Cette vue améliore la portabilité des modèles et évite les requêtes de métadonnées exécutées lorsque des calculs sont effectués directement dans les sujets de requête de source de données.

- Vue dimensionnelle

Contient les dimensions régulières, les dimensions de mesure et les relations de portée qui ont été créées dans Framework Manager.

- Vue d'analyse

Contient la partie du modèle qui est visible dans les studios pour le pack (dimensionnel) d'analyse.

Les raccourcis vers les dimensions ordinaires et de mesure de la vue Dimensions sont regroupés dans des schémas en étoile et placés à racine du modèle dans différents dossiers, un pour chaque domaine métier. Ils sont clairement signalés avec la mention "(analyse)" à la fin du nom. Ils ne sont pas situés dans un espace-noms distinct, car cela ajouterait un autre niveau dans l'arborescence des métadonnées dans les studios.

- Vue de requête

Contient la partie du modèle qui est visible dans les studios pour le pack de requêtes et de reporting.

Les raccourcis vers les sujets de requête de modèle de la vue Dimensions sont regroupés dans des schémas en étoile et placés dans différents dossiers, un pour chaque domaine métier. Ils sont clairement signalés avec la mention "(requête)" à la fin du nom à la racine du modèle.

En outre, l'exemple de modèle contient des packs, un pour l'analyse et un pour les requêtes et le reporting. Vous ne pouvez pas utiliser les packs de requêtes et de reporting dans IBM Cognos Analysis Studio.

## Modèle Ventes Vacances et Aventure

Ce modèle contient des informations sur l'analyse des ventes de la société fictive Vacances et Aventure. Ce modèle accède à un système transactionnel.

Cet exemple de modèle est situé dans `c10_location\webcontent\samples\models\great_outdoors_sales`.

L'exemple de modèle contient les vues suivantes :

- Vue de base de données  
Contient les sujets de requête qui ont été importés à partir de la source de données. Comme il s'agit d'une source de données transactionnelle, la plupart des tables de faits ne possèdent pas de clés. Dans certains cas, nous avons utilisé des clés multiparties. Dans d'autres cas, des sujets de requête de modèle ont été ajoutés pour calculer les clés et résoudre la cardinalité ambiguë.  
Toutes les jointures et tous les déterminants sont définis dans cette vue.
- Vue métier  
Contient des sujets de requête de modèle et des raccourcis de référence qui représentent les données en termes de besoins métier ou d'application. Les calculs, y compris ceux pour le langage, sont définis dans cette vue.
- Vue dimensionnelle  
Contient les dimensions régulières, les dimensions de mesure et les relations de portée. Les dimensions sont basés sur les sujets de requête de modèle de la vue métier.  
C'est également dans cette vue que les sujets de requête sont renommés si nécessaire.
- Vue d'analyse  
Contient la partie du modèle qui est visible dans les studios pour le pack (dimensionnel) d'analyse.  
Les raccourcis vers les dimensions ordinaires et de mesure de la vue Dimensions sont regroupés dans des schémas en étoile et placés à racine du modèle dans différents dossiers, un pour chaque domaine métier. Ils sont clairement signalés avec la mention "(analyse)" à la fin du nom. Ils ne sont pas situés dans un espace-noms distinct, car cela ajouterait un autre niveau dans l'arborescence des métadonnées dans les studios.
- Vue de requête  
Contient la partie du modèle qui est visible dans les studios pour le pack de requêtes et de reporting.  
Les raccourcis vers les sujets de requête de modèle de la vue Dimensions sont regroupés dans des schémas en étoile et placés dans différents dossiers, un pour chaque domaine métier. Ils sont clairement signalés avec la mention "(requête)" à la fin du nom à la racine du modèle.

En outre, l'exemple de modèle contient des packs, un pour l'analyse et un pour les requêtes et le reporting. Vous ne pouvez pas utiliser les packs de requêtes et de reporting dans IBM Cognos Analysis Studio.



---

## Chapitre 3. Model Design Accelerator

Model Design Accelerator est une extension d'IBM Cognos Framework Manager qui simplifie la création de modèles de schéma en étoile relationnel. Celle-ci permet modéliser un schéma en étoile relationnel de table de faits unique respectant des pratiques de modélisation éprouvées. Les modélisateurs débutants peuvent créer facilement des modèles sans pour cela posséder beaucoup d'expérience ni avoir suivi une formation approfondie. Les modélisateurs expérimentés peuvent réduire le temps global nécessaire à la création d'un modèle. Model Design Accelerator évalue chaque étape de conception dans le but d'identifier les problèmes potentiels et de vous aider à les résoudre.

Avant de commencer à concevoir un modèle, vous devez comprendre le problème de génération de rapports que vous tentez de résoudre et savoir de quelles données vous disposez pour cela.

Pour commencer à utiliser Model Design Accelerator, procédez comme suit :

- Créez un projet et sélectionnez une source de données.
- Explorez l'interface utilisateur de Model Design Accelerator.
- Créez un modèle de schéma en étoile.
- Découvrez comment gérer votre modèle de schéma en étoile.

Model Design Accelerator comprend plusieurs vues et boîtes de dialogue qui vous aideront à créer votre modèle.

- Arborescence de l'explorateur
- Diagramme d'explorateur
- Model Accelerator
- Diagramme de sujet de requête
- Mode d'édition de relation
- Avertissement de modèle
- Options

Pour accéder aux menus contextuels de chaque vue, cliquez avec le bouton droit dans la vue. La boîte de dialogue **Options** est disponible à partir de tous les menus contextuels.







---

### Arborescence d'explorateur

L'**arborescence d'explorateur** vous permet d'afficher les objets que vous avez sélectionnés à l'aide de l'assistant de métadonnées. Les objets de source de données sont affichés dans une vue hiérarchique, semblable à d'autres systèmes de fichiers.


Pour afficher une vue graphique de votre source de données, utilisez le **diagramme d'explorateur**. Si la source de données comporte un grand nombre d'objets, il sera peut-être plus facile de retrouver un objet à l'aide de l'**arborescence d'explorateur**.

L'**arborescence d'explorateur** utilise les icônes suivantes pour représenter des objets.

Icône	Objet
	Projet
	Table dans la source de données
	Colonne dans la source de données La propriété de type de données de la colonne l'identifie en tant que mesure.
	Colonne dans la source de données Les propriétés de la colonne l'identifient en tant que clé.
	Colonne dans la source de données Les propriétés de la colonne ne l'identifient pas en tant que mesure ni en tant que clé.
	Source de données

## Diagramme d'explorateur

Le **diagramme d'explorateur** affiche une vue graphique de vos métadonnées de source de données. Utilisez le **diagramme d'explorateur** pour explorer vos métadonnées et afficher les relations entre les objets.

Pour accéder au **diagramme d'explorateur**, sélectionnez un ou plusieurs objets dans l'**arborescence d'explorateur**. Ensuite, cliquez sur l'icône **Diagramme d'explorateur**  située au-dessus de l'**arborescence d'explorateur**.

Dans la fenêtre **Diagramme d'explorateur**, vous pouvez exécuter les actions suivantes :

- Afficher des objets et des relations.
- Modifier la présentation des objets et afficher ces derniers en étoile ou dans une arborescence standard.
- Créer un schéma en étoile préliminaire basé sur une table contenant des faits.
- Effectuer un zoom avant ou un zoom arrière.
- Modifier les paramètres des diagrammes.

Il existe deux façons d'utiliser le **diagramme d'explorateur**. Dans l'**arborescence d'explorateur**, vous pouvez sélectionner un sous-ensemble d'objets qui vous intéresse et l'ajouter au **diagramme d'explorateur**. Dans l'**arborescence d'explorateur**, vous pouvez également ajouter tous les objets au **diagramme d'explorateur**. Vous pouvez alors conserver les objets qui vous intéressent et supprimer les autres objets. Utilisez les éléments de menu **Afficher les tables liées** et **Retirer les tables** pour créer une vue ne contenant que les objets que vous souhaitez explorer. Le diagramme est une vue accessible en lecture seule qui contient vos métadonnées de source de données. Lorsque vous ajoutez ou supprimez les tables, vous ne modifiez que le diagramme. Vous ne modifiez pas la source de données.

Utilisez les options de menu du diagramme pour explorer les relations avec d'autres objets. Pour accéder aux options de menu du diagramme, sélectionnez un objet dans le **diagramme d'explorateur** et cliquez dessus avec le bouton droit de la souris ou utilisez la barre d'outils. Cliquez avec le bouton droit de la souris sur

une table pour sélectionner l'option permettant d'afficher le diagramme dans une présentation en étoile ou dans une arborescence autour de cette table. Vous pouvez également accéder à certaines options de menu du diagramme à l'aide des icônes.

Vous pouvez créer un schéma en étoile préliminaire en sélectionnant n'importe quelle table dans le **diagramme d'explorateur**. Cliquez dessus avec le bouton droit de la souris et sélectionnez **Générer un schéma en étoile à partir de cette table** pour créer une présentation de schéma en étoile complète.

Vous pouvez ancrer le **diagramme d'explorateur** sur l'espace de travail en cliquant sur le bouton d'ancrage. Dans la fenêtre ancrée, vous avez la possibilité de diviser votre écran horizontalement ou verticalement.

Utiliser la boîte de dialogue **Options** pour personnaliser les paramètres de diagramme par défaut. Par exemple, si vos tables contiennent de nombreuses colonnes ou utilisent des noms longs, les données présentées dans les diagrammes peuvent ne pas s'afficher complètement. Ajustez les options de taille de table maximale sous l'onglet **Général** pour faire en sorte que toutes vos données soient affichées. Utilisez l'onglet **Couleurs** pour modifier les couleurs par défaut utilisées pour identifier les caractéristiques du diagramme.

---

## Espace de travail Model Accelerator

Une fois que vous avez créé un projet, vous pouvez commencer à travailler dans une session Model Design Accelerator. **Model Accelerator** correspond à l'espace de travail graphique dans lequel vous pouvez générer votre schéma en étoile.

Pour générer votre schéma en étoile, concevez le sujet de requête de fait en utilisant les mesures que vous souhaitez inclure dans vos rapports. Ensuite, générez des sujets de requête de modèle afin de fournir du contexte et décrire ces mesures. Les lignes qui relient les sujets de requête indiquent que des relations existent. Utilisez le **diagramme de sujet de requête** pour afficher les tables de source de données qui ont été utilisées pour créer le sujet de requête sélectionné.

Model Design Accelerator évalue chaque étape de conception par rapport à des pratiques de modélisation éprouvées dans le but d'identifier les problèmes potentiels et de vous aider à les résoudre. Si un problème potentiel est généré par vos actions, la vue **Avertissement de modèle** s'affiche.

Une fois que vous avez terminé votre schéma en étoile, générez un modèle. Vous pouvez affiner le modèle dans Framework Manager. Créez un pack et publiez le modèle dans IBM Cognos Connection pour commencer à créer des rapports dans les studios.

## Diagramme de sujet de requête

Utilisez le **diagramme de sujet de requête** pour afficher les tables de source de données qui ont été utilisées pour créer le schéma en étoile. Ces tables sont importées dans Framework Manager en tant que partie de votre modèle généré. Pour accéder au **diagramme de sujet de requête**, cliquez deux fois sur un sujet de requête dans l'espace de travail **Model Accelerator**.

Dans la fenêtre **Diagramme de sujet de requête**, vous pouvez exécuter les actions suivantes :

- Afficher les tables de source de données qui ont été utilisées pour créer le sujet de requête sélectionné.

- Créer, modifier, supprimer ou remplacer des relations à l'aide du **mode d'édition de relation**.
- Modifier les paramètres des diagrammes à l'aide de la boîte de dialogue **Options**.

Utiliser la boîte de dialogue **Options** pour personnaliser les paramètres de diagramme par défaut. Par exemple, si vos tables contiennent de nombreuses colonnes ou utilisent des noms longs, les données présentées dans les diagrammes peuvent ne pas s'afficher complètement. Ajustez les options de taille de table maximale dans la boîte de dialogue **Options** pour faire en sorte que toutes vos données soient affichées. Une autre option contrôle les couleurs par défaut qui sont utilisés pour identifier les caractéristiques du diagramme.

## Utilisation du mode d'édition de relation

Le **mode d'édition de relation** permet de créer, modifier, supprimer ou remplacer des jointures de modèle.

Vous pouvez accéder au **mode d'édition de relation** de deux manières :

- Si Model Design Accelerator détecte un problème potentiel, la fenêtre **Avertissement de modèle** s'affiche. L'option de **nouveau traçage manuel des jointures entre les tables** active la fonctionnalité du mode d'édition de relation à partir de la vue **Avertissement de modèle**.
- Vous pouvez cliquer deux fois sur un sujet de requête dans l'espace de travail **Model Accelerator** pour accéder au **diagramme de sujet de requête**. Vous pouvez alors cliquer sur **Accéder au mode de création de relation**.

### Procédure

1. Sélectionnez deux tables à joindre.

Si vous ne pouvez pas créer de jointure directe entre deux tables, utilisez des tables intermédiaires. Pour ajouter des tables intermédiaires, faites glisser les tables depuis de l'**explorateur d'arborescence** vers le **diagramme de sujet de requête**. Créez ensuite les jointures requises en reliant les tables par le biais des tables intermédiaires.

**Astuce** : Une table intermédiaire a une nomenclature en pointillés.


2. Cliquez sur l'icône **Créer un modèle de relation** .

La boîte de dialogue **Modifier la relation** s'affiche. Une nouvelle ligne de jointure apparaît.

3. Sélectionnez une colonne dans chaque table pour créer la relation.
4. Sélectionnez la cardinalité de relation appropriée.
5. Cliquez sur **OK** deux fois pour revenir dans l'espace de travail **Model Accelerator**.

Si votre action aboutit à un modèle non valide, la vue **Avertissement de modèle** s'affiche.

### Résultats

Vous pouvez également depuis le **mode d'édition de relation** sélectionner les colonnes à joindre, puis cliquer sur l'icône **Créer un modèle de relation** . La boîte de dialogue **Modifier la relation** apparaît avec la jointure affichée entre les colonnes sélectionnées.



## Vue Avertissement de modèle

La vue **Avertissement de modèle** s'affiche lorsque Model Design Accelerator détecte un problème potentiel. Pour vous aider à générer un modèle valide, Model Design Accelerator évalue chacune de vos actions. La vue **Avertissement de modèle** décrit l'action qui est à l'origine d'un problème et, lorsque cela s'applique, fournit des options permettant de résoudre ce problème.

Certaines options présentées sont effectuées automatiquement lorsque vous les sélectionnez et que vous cliquez sur **OK**. Pour d'autres options, vous devez apporter les modifications manuellement, soit dans Model Design Accelerator, soit dans le modèle généré dans Framework Manager.

Etant donné que Model Design Accelerator n'est pas toujours en mesure de déterminer ce que vous cherchez à faire au moyen de vos actions, vous devez parfois résoudre un problème en créant des jointures manuelles. Lorsqu'elle est disponible, l'option de **nouveau traçage manuel des jointures entre les tables** active la fonctionnalité du mode d'édition de relation à partir de la vue Avertissement de modèle.

Lorsque vous sélectionnez l'option de **nouveau traçage manuel des jointures entre les tables**, des icônes de jointure s'affichent dans l'angle supérieur gauche de la vue Avertissement de modèle. Vous pouvez créer autant de jointures manuelles que vous le souhaitez pour résoudre le problème identifié.

Pour plus d'informations, voir «Utilisation du mode d'édition de relation», à la page 30.

S'il existe une série d'options de modélisation dépendantes, cliquez sur **Annuler** pour annuler votre dernière action. S'il n'existe aucune option de modélisation, cliquez sur **Annuler** pour annuler la dernière action et revenir à l'espace de travail **Model Accelerator**.

---

## Modification des paramètres de diagrammes

Vous pouvez modifier un ou plusieurs paramètres des vues de diagramme à l'aide de la boîte de dialogue **Options**. Il existe trois vues de diagramme dans Model Design Accelerator : l'espace de travail **Model Accelerator**, le **diagramme d'explorateur** et le **diagramme de sujet de requête**.

### Procédure

1. Dans l'une des vues de diagramme, cliquez avec le bouton droit de la souris pour accéder au menu contextuel. Sélectionnez **Options** et cliquez sur l'onglet **Général**.
2. Sélectionnez le type de notation pour représenter les relations.  
Vous pouvez utiliser la notation Crowsfoot simplifiée, Crowsfoot standard ou de cardinalité numérique. Par défaut, Model Design Accelerator utilise la notation Crowsfoot simplifiée qui fournit une représentation illustrée de la relation.
3. Définissez une taille de table maximale pour les diagrammes d'explorateur et de sujet de requête.
4. Définissez une taille de table maximale pour l'espace de travail Model Accelerator.
5. Indiquez si vous souhaitez qu'une description s'affiche en regard de chaque icône d'outil.

6. Indiquez si vous souhaitez que l'écran d'introduction s'affiche chaque fois que vous démarrez **Model Design Accelerator**.
7. Cliquez sur l'onglet **Couleurs** pour modifier les couleurs des objets de diagramme.  
Des couleurs par défaut sont affectées aux différentes caractéristiques des objets de diagramme. Modifiez les couleurs pour personnaliser l'affichage.  
**Afficher les couleurs d'arrière-plan** indique que les modifications apportées dans la fenêtre active vont entraîner des modifications dans la conception de schéma en étoile. Cela vous permet d'identifier facilement les fenêtres utilisées pour l'édition et celles utilisées uniquement pour l'affichage.
8. Cliquez sur **OK**.

---

## Création d'un projet

Dans Model Design Accelerator, vous travaillez dans dans le contexte d'un projet.

### Procédure

1. Dans la page **Bienvenue** d'IBM Cognos Framework Manager, cliquez sur **Créer un projet à l'aide de Model Design Accelerator**.  
Si vous êtes dans Framework Manager, vous pouvez sélectionner **Exécuter Model Design Accelerator** dans le menu **Outils**. Dans ce cas, un nouveau projet n'est pas créé. Le schéma en étoile est associé au projet ouvert et le modèle généré est ajouté à ce projet.
2. Dans la page **Nouveau projet**, spécifiez un nom et un emplacement pour le projet et cliquez sur **OK**.
3. Dans la page **Sélectionner la langue**, cliquez sur la langue de conception du projet.
4. Cliquez sur **OK** pour sélectionner la langue de conception.  
Vous ne pouvez pas modifier la langue de conception mais vous pouvez ajouter d'autres langues dans Framework Manager.  
L'**assistant de métadonnées** s'affiche.
5. Pour sélectionner votre source de données, suivez les instructions de l'**assistant de métadonnées** :
  - Sélectionnez une source de données et cliquez sur **Suivant**.  
Si la source de données souhaitée n'est pas répertoriée, vous devez d'abord la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.  
Vous ne pouvez importer qu'à partir d'une seule source de données à la fois. Si vous souhaitez importer à partir de plusieurs sources de données, vous devez effectuer plusieurs importations.
  - Cochez les cases correspondant aux objets que vous souhaitez utiliser.  
Vous pouvez sélectionner le nombre d'objets souhaité. Toutes les métadonnées sélectionnées sont disponibles pour que vous les utilisiez dans votre conception de schéma en étoile. Seuls les objets requis pour le modèle sont importés dans Framework Manager lorsque vous générez le modèle.
  - Cliquez sur **Continuer** pour ouvrir Model Design Accelerator.  
Un écran d'introduction qui vous donne un aperçu des étapes de la conception peut s'afficher. Fermez l'écran d'introduction pour commencer à utiliser Model Design Accelerator.
6. Créer votre modèle de schéma en étoile à l'aide des objets que vous avez sélectionnés dans la source de données.

7. Effectuez l'une des opérations suivantes :
    - Cliquez sur **Enregistrer** pour enregistrer le contenu de la session et continuer à travailler.
    - Cliquez sur **Fermer** pour enregistrer le contenu de la session et ouvrir Framework Manager.
    - Cliquez sur **Générer un modèle** pour générer un modèle basé sur votre conception de schéma en étoile et ouvrir Framework Manager.
- Vous pouvez enregistrer et fermer votre conception aussi souvent que vous le souhaitez sans générer de modèle. Pour plus d'informations, voir «Schéma en étoile»

---

## Schéma en étoile

La création d'un schéma en étoile à l'aide de Model Design Accelerator est un processus en trois étapes :

- Créez le sujet de requête de fait avec les mesures que vous souhaitez inclure dans vos rapports.
- Créez les sujets de requête associés pour donner du contexte aux mesures.
- Générez le modèle.

A tout moment, vous pouvez survoler un objet avec le pointeur de la souris pour en afficher les propriétés. Les propriétés indiquent l'origine de l'objet dans la source de données. La vue du **diagramme d'explorateur** affiche également l'élément lorsqu'il est utilisé dans le schéma en étoile. Vous pouvez cliquer avec le bouton droit sur une jointure pour afficher la **boîte de dialogue des relations**.

Il est recommandé de sauvegarder la conception au fur et à mesure. Si vous quittez Model Design Accelerator et que vous y accédez de nouveau, vous pouvez reprendre votre travail sur le schéma en étoile que vous avez sauvegardé.

**Remarque :** Lorsque vous accédez de nouveau à Model Design Accelerator, il s'ouvre dans votre dernière session de travail. Vous ne pouvez pas accéder à d'autres projets à partir de Model Design Accelerator.

### Création du sujet de requête de fait

Utilisez l'**arborescence d'explorateur** pour explorer les tables et les colonnes de votre source de données. Sélectionnez les mesures que vous souhaitez inclure dans vos rapports et ajoutez-les au sujet de requête de fait dans l'espace de travail Model Accelerator. Les mesures sont les colonnes numériques d'une table ou d'une vue de base de données, qui peuvent être regroupées ou agrégées. Coût de production ou Quantité sont des exemples de mesure. Model Design Accelerator contrôle les clés requises et les inclut automatiquement dans le modèle. Il n'est pas obligatoire d'ajouter des clés. Sélectionnez uniquement les mesures que les auteurs du rapport doivent pouvoir visualiser dans votre pack.

Sélectionnez des mesures dans l'**arborescence d'explorateur**, puis faites-les glisser sur le sujet de requête de fait dans l'espace de travail **Model Accelerator**. Vous pouvez également effectuer vos sélections à partir du **diagramme d'explorateur**.

Toutes les mesures en cours d'ajout doivent provenir de la même table dans votre source de données. Si vous tentez d'ajouter des mesures provenant de plusieurs tables ou d'ajouter des éléments de requête qui ne sont pas des mesures, la boîte de dialogue **Avertissement de modèle** s'affiche. Cela est dû au fait que Model

Design Accelerator prend en charge uniquement un seul modèle de schéma de table de faits en étoile.

## Création des sujets de requête associés

Utilisez des objets depuis l'**arborescence d'explorateur** pour générer les sujets de requête dans le modèle de schéma en étoile. Sélectionnez des colonnes à partir de l'**arborescence d'explorateur** et ajoutez-les aux sujets de requête appropriés dans l'espace de travail Model Accelerator. Vous pouvez ajouter ou supprimer les colonnes ou les sujets de requête de l'espace de travail en fonction des besoins.

Pour vous aider à créer les sujets de requête, utilisez le **diagramme d'explorateur** et examinez les métadonnées des tables et des vues que vous avez sélectionnées dans la source de données. Vous pouvez créer des sujets de requête à partir de la **vue d'explorateur** ou du **diagramme d'explorateur**.

Dans le **diagramme d'explorateur**, vous pouvez sélectionner un ou plusieurs sujets de requête et visualiser les relations.

## Génération du modèle

Lorsque votre schéma en étoile est terminé ou prêt pour le test, cliquez sur **Générer un modèle** pour créer le modèle et accédez à IBM Cognos Framework Manager.

Les tests effectués par **Model Advisor** sont appliqués au modèle obtenu. Sous l'onglet **Vérifier les résultats du modèle**, examinez les problèmes qui ont été identifiés. Chaque problème est décrit et tous les objets impactés par chacun de ces problèmes sont répertoriés. Pour plus d'informations sur Model Advisor, voir «Analyse de modèles», à la page 190.

Le modèle de schéma en étoile que vous avez créé est généralement destiné à contenir uniquement un sous-ensemble des éléments de requête provenant de la source de données. Lorsque vous générez un modèle, seules les métadonnées nécessaires à la prise en charge de votre modèle de schéma en étoile sont importées dans Framework Manager.

Chaque fois que vous générez un modèle, de nouveaux espaces-noms sont créés dans le projet Framework Manager ouvert. La **vue physique** contient uniquement les tables qui ont été importées pour permettre la prise en charge du modèle de schéma en étoile. La **vue métier** contient des sujets de requête de modèle comportant uniquement les colonnes que vous avez sélectionnées lors de la création de sujets de requête. Il s'agit des objets que vous avez créés dans Model Design Accelerator. La **vue de présentation** contient une collection de raccourcis qui font référence aux sujets de requête de modèle présents dans la vue métier.

Framework Manager vous permet d'améliorer le modèle en y ajoutant des calculs, des filtres, d'autres langues et des structures dimensionnelles. Cependant, chaque fois que vous générez un nouveau modèle, de nouveaux espaces-noms sont créés et les améliorations que vous avez réalisées ne sont pas appliquées à ces nouveaux espaces-noms.

Vous pouvez revenir à Model Design Accelerator à tout moment pour éditer votre schéma en étoile. Vous devez publier le modèle dans Framework Manager afin de pouvoir l'utiliser dans les studios d'IBM Cognos.

---

## Gestion du schéma en étoile

L'une des méthodes visant à concevoir un schéma en étoile consiste à créer un modèle autonome. Pour ce faire, créez un projet, créez un schéma en étoile dans Model Design Accelerator, générez et enregistrez un modèle dans le projet IBM Cognos Framework Manager.

Vous pouvez enregistrer et fermer votre conception de modèle de schéma en étoile aussi souvent que vous le souhaitez sans générer de modèle. Après avoir généré un modèle, vous pouvez revenir à Model Design Accelerator en sélectionnant **Exécuter Model Design Accelerator** dans le menu **Outils**. Vous pouvez exécuter l'une des actions suivantes :

- Editer le schéma en étoile en cours, le cas échéant.
- Créer un nouveau schéma en étoile en utilisant les mêmes métadonnées de source de données importées.

Les métadonnées Model Design Accelerator sont enregistrées dans le fichier mda\_metadata.xml dans le dossier de projet. Vous pouvez déplacer le schéma en étoile vers un autre projet en copiant le fichier mda\_metadata.xml dans un autre dossier de projet.

Chaque fois que vous générez un modèle, des espaces-noms sont créés dans le projet Framework Manager ouvert. La **vue physique** contient uniquement les tables qui ont été nécessaires pour permettre la prise en charge du modèle de schéma en étoile. La **vue métier** contient des sujets de requête de modèle comportant uniquement les colonnes que vous avez sélectionnées lors de la création des sujets de requête. Il s'agit des objets que vous avez créés dans Model Design Accelerator. La **vue de présentation** contient une série de raccourcis vers les sujets de requête de modèle que vous avez créés dans la vue métier.

Pour chaque modèle généré par la suite, un nouvel ensemble d'espaces-noms est créé dans le projet Framework Manager. Un numéro est ajouté à chaque nom d'espace-noms pour le distinguer de la version précédente. Par exemple, le premier modèle contient un espace-noms intitulé Vue physique. Le second modèle contiendra un espace-noms intitulé Vue physique1.

Lorsque vous avez terminé votre conception de schéma en étoile, vous pouvez améliorer le modèle obtenu dans Framework Manager en y ajoutant des calculs, des filtres, d'autres langues et des structures dimensionnelles. Si vous modifiez le schéma en étoile et que vous régénérez un modèle, votre conception d'origine et les améliorations que vous avez réalisées ne sont pas écrasées.



---

## Chapitre 4. Importation de métadonnées à partir de sources de données

Avant d'effectuer une modélisation dans IBM Cognos Framework Manager, vous devez importer des métadonnées. Vous pouvez importer des métadonnées à partir d'une grande variété de sources de données, relationnelles et dimensionnelles. Vous pouvez effectuer l'importation dans un nouveau modèle ou un modèle existant. Avant l'importation, il est important de vérifier que la source de données contient bien les données et les métadonnées qui répondent à vos besoins de génération de rapports.

Pour importer des métadonnées, procédez comme suit :

### Procédure

1. Définissez des sources de données.
2. Définissez la sécurité des sources de données.
3. Renseignez-vous sur les types de connexion de source de données.
4. Apprenez à gérer les connexions de source de données.
5. Créez une connexion de source de données.
6. Importez les métadonnées.

---

### Sources de données

Pour pouvoir créer des modèles et importer des métadonnées, vous devez définir des sources de données. Une connexion de source de données fournit les informations dont IBM Cognos BI a besoin pour se connecter à une base de données.

Chaque source de données peut contenir une ou plusieurs connexions physiques à des bases de données. La connexion de source de données indique les paramètres nécessaires pour se connecter à la base de données, tels que l'emplacement de la base de données et la durée du délai d'attente. Une connexion peut inclure des informations sur les données d'identification et des codes d'accès.

Vous pouvez sécuriser des sources de données à l'aide de l'authentification IBM Cognos. L'authentification IBM Cognos respecte la sécurité qui est définie aussi dans la source de données. Vous pouvez créer des codes d'accès de source de données pour isoler des utilisateurs finaux du processus de connexion à la base de données. Le code d'accès stocke l'ID utilisateur et le mot de passe requis pour accéder à la base de données. Vous pouvez également déployer des sources de données.

Pour plus d'informations sur les connexions de source de données, voir le *Guide d'administration et de sécurité IBM Cognos*.

---

## Sécurité des sources de données

Vous pouvez définir un système de sécurité pour les sources de données à l'aide de l'authentification IBM Cognos ou du mécanisme de sécurité spécifique à une source de données. La définition de l'authentification IBM Cognos pour une source de données ne remplace pas les règles de sécurité du fournisseur de base de données.

Selon la source de données utilisée, une ou plusieurs des méthodes d'authentification IBM Cognos suivantes sont disponibles :

- Aucune authentification  
IBM Cognos BI se connecte à la source de données sans fournir de données d'identification de code d'accès.
- Données d'identification du service IBM Cognos  
IBM Cognos BI se connecte à la source de données à l'aide de la connexion spécifiée pour le service IBM Cognos. Les utilisateurs n'ont pas besoin de codes d'accès de base de données individuelles. Pour les environnements de production par contre, les codes d'accès de base de données individuelles sont généralement plus adaptés.
- Espace-noms externe  
IBM Cognos BI se connecte à la source de données à l'aide des mêmes données d'identification que celles utilisées pour l'authentification auprès de l'espace-noms externe spécifié. L'espace-noms spécifié doit être actif, les utilisateurs doivent y être connectés avant d'accéder à la source de données et les données d'identification utilisées pour l'authentification de l'espace-noms doivent permettre l'authentification de la source de données.

Toutes les sources de données prennent également en charge les codes d'accès à la source de données définis pour le groupe Tous ou pour des utilisateurs, des groupes ou des rôles individuels. Si la source de données nécessite un code d'accès, mais que vous n'avez pas accès à un code d'accès, vous êtes invité à vous connecter chaque fois que vous accédez à celle-ci.

IBM Cognos BI respecte également la sécurité définie pour la source de données. Par exemple, pour les cubes IBM Cognos, la sécurité peut être définie au niveau du cube. Pour les sources de données Microsoft Analysis Services, la sécurité peut être définie à l'aide de rôles de cubes.

---

## Types de connexions de source de données

IBM Cognos prend en charge différents types de sources de données. Les informations de connexion de source de données peuvent varier selon le type de source de données que vous utilisez.

Pour plus d'informations sur les connexions pour des sources de données spécifiques, voir le *Guide d'administration et de sécurité IBM Cognos*. Si vous avez besoin d'informations complémentaires sur les paramètres de connexion à votre source de données, reportez-vous à la documentation du fournisseur de la source de données que vous utilisez.


Dans un modèle Framework Manager, les catalogues et les schémas sont des propriétés associées à un objet de source de données. Si elles sont présentes, ces propriétés qualifient les objets générés dans une instruction SQL lors de l'exécution.



Lorsque vous basculez vers la source de données sur laquelle un modèle est mappé, des niveaux de qualification sont souvent supprimés. Le système de gestion de base de données relationnelle (SGBDR) applique des recherches d'espace-noms appropriées lors de la préparation des instructions SQL reçues. Selon le fournisseur, le système de gestion de base de données relationnelle peut explorer au maximum deux emplacements pour résoudre une référence à un objet. Une erreur se produit si une référence ne peut pas être résolue.

IBM Cognos Framework Manager préserve les noms des tables et des colonnes tels qu'ils apparaissent dans le système de gestion de base de données relationnelle. Ces noms s'affichent dans les instructions SQL Cognos générées. Cognos SQL utilise des guillemets pour les noms afin de conserver la casse et d'éviter les conflits avec les caractères ou les mots clés spéciaux.

## Métadonnées natives

IBM Cognos prend en charge les sources de données OLAP ainsi que les sources de données relationnelles. Le terme "métadonnées natives" fait référence à des objets tels que des modèles, des packs et des requêtes qui sont basés sur une source de données OLAP. Un espace-noms qui contient des métadonnées natives utilise l'icône  pour signaler qu'il est différent des espaces-noms contenant d'autres types de métadonnées.

Les sources de données OLAP sont des sources de données riches de métadonnées. La modélisation explicite n'est pas activée pour ces sources de données dans Framework Manager et le pack est publié directement sur le portail. Pour plus d'informations, voir «Publication de packs basés sur des sources de données OLAP», à la page 271.

Les niveaux sont créés à l'aide des noms de génération de noms dans les libellés. Vous pouvez changer le mode d'appellation des niveaux en modifiant les paramètres de génération de dimension dans l'application ayant généré le cube. Pour plus d'informations, voir la documentation fournisseur.

Les sources de données relationnelles nécessitent la modélisation dimensionnelle pour fonctionner dans IBM Cognos Analysis Studio et être utilisées avec les fonctions d'accès aux détails dans les autres studios. Pour plus d'informations sur la modélisation dimensionnelle, voir «Dimensions», à la page 106.

Si vous avez installé des composants IBM Cognos sur des serveurs UNIX, localisez la source de données basée sur des fichiers sur un serveur UNIX. Utilisez ensuite un chemin d'accès UNIX, tel que /servername/cubes/Great Outdoors Company.mdc, pour accéder au fichier. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.

Les packs composés contiennent des métadonnées relationnelles et des métadonnées OLAP.

---

## Connexions de source de données

Vous pouvez personnaliser des connexions de source de données afin de répondre aux besoins des utilisateurs.

## Connexions de source de données multiples

Si vous avez accès à plusieurs connexions de source de données dans une source de données, vous êtes invité à en sélectionner une lorsque vous ouvrez un projet IBM Cognos Framework Manager. Vous pouvez utiliser plusieurs connexions de source de données dans une seule source de données pour faciliter la migration d'un environnement à un autre et préserver l'intégrité d'un projet.

Par exemple, vous pouvez faire appel à plusieurs connexions de source de données pour travailler avec des métadonnées issues d'une source de données de test. Créez un projet à l'aide de la connexion de source de données GoSales. Créez et modifiez les objets que vous souhaitez inclure dans le projet et effectuez un test pour vous assurer que le projet est modélisé comme vous le souhaitez. Après avoir fermé la session, puis rouvert le projet Framework Manager, vous pouvez sélectionner la connexion de source de données de production. Lorsque vous publiez le pack sur le serveur IBM Cognos, les utilisateurs sélectionnent la connexion de source de données qu'ils souhaitent utiliser dans leur rapport.

Plusieurs connexions à la même source de données doivent être définies dans IBM Cognos Connection. Si vous souhaitez prendre en charge plusieurs connexions pour chaque source de données, effacez les noms de catalogue et de schéma de source de données et créez une connexion pour chaque base de données dans IBM Cognos Connection.

**Remarque :** Si vous utilisez plusieurs cubes contenant des métadonnées dissemblables, utilisez des sources de données distinctes pour chaque cube. Pour que vous puissiez développer un pack OLAP dans les studios, le nom interne de ces cubes doit être le même. Si vous souhaitez exécuter des rapports enregistrés utilisant des connexions de source de données différentes, le nom de cube, ainsi que les noms de dimension, de hiérarchie, de niveau et d'attribut doivent être les mêmes dans les cubes. Si vous utilisez une source de données unique avec une connexion distincte pour chaque cube, les noms internes de tous les cubes doivent être les mêmes.

Pour plus d'informations sur les connexions de source de données, voir le *Guide d'administration et de sécurité IBM Cognos*.

## Niveaux d'isolement

Le niveau d'isolement indique la manière dont les transactions qui modifient la base de données sont gérées. Par défaut, la passerelle d'objet par défaut est utilisée. Tous les types de base de données ne prennent pas en charge chaque niveau d'isolement. Certains fournisseurs de bases de données utilisent d'autres noms pour les niveaux d'isolement.

Les requêtes exécutées par des rapports et des analyses sont destinées à être des opérations en lecture seule. Les requêtes s'exécutent avec une unité de travail au niveau de la source de données connue sous le nom de transaction, avec un niveau d'isolement par défaut ou défini par l'administrateur. Les auteurs de rapports ne doivent pas considérer que les requêtes qui exécutent des procédures stockées valident les données écrites par ces procédures. Dans certains environnements, les modifications apportées par une procédure peuvent être validées en raison des fonctionnalités de la base de données. Une procédure stockée qui est marquée pour un accès en écriture dans Framework Manager valide des modifications, mais ne peut être utilisée que par Event Studio.

Si vous avez besoin que certaines requêtes s'exécutent avec différents niveaux d'isolement, vous devez définir différentes connexions à la base de données.

Pour les sources de données OLAP, y compris SAP BW, l'unité de travail de la transaction est accessible en lecture seule.

Les niveaux d'isolement suivants sont présentés par ordre croissant d'isolement :

- Lecture non validée  
Les modifications apportées par d'autres transactions sont immédiatement disponibles pour une transaction.

*Tableau 1. Types de base de données à lecture non validée et niveaux d'isolement équivalents*

Type de base de données	Niveau d'isolement équivalent
Oracle	Sans objet
DB2	Lecture non validée
Microsoft SQL Server	Lecture non validée
Sybase Adaptive Server Enterprise	Lecture non validée
Informix	Lecture de pages modifiées

- Lecture validée  
Une transaction ne peut accéder qu'aux lignes validées par d'autres transactions.

*Tableau 2. Types de base de données à lecture validée et niveaux d'isolement équivalents*

Type de base de données	Niveau d'isolement équivalent
Oracle	Lecture validée
DB2	Lecture non reproductible
Microsoft SQL Server	Lecture validée
Sybase Adaptive Server Enterprise	Lecture validée
Informix	Lecture validée

- Lecture non reproductible  
Les autres transactions ne peuvent pas mettre à jour la ligne dans laquelle figure une transaction.

*Tableau 3. Types de base de données à lecture non reproductible et niveaux d'isolement équivalents*

Type de base de données	Niveau d'isolement équivalent
Oracle	Sans objet

Tableau 3. Types de base de données à lecture non reproductible et niveaux d'isolement équivalents (suite)

Type de base de données	Niveau d'isolement équivalent
DB2	Sans objet
Microsoft SQL Server	Sans objet
Sybase Adaptive Server Enterprise	Sans objet
Informix	Lecture non reproductible

- Lecture reproductible

Les lignes sélectionnées ou mises à jour par une transaction ne peuvent pas être modifiées par une autre transaction tant que la transaction n'est pas terminée.

Tableau 4. Types de base de données à lecture reproductible et niveaux d'isolement équivalents

Type de base de données	Niveau d'isolement équivalent
Oracle	Sans objet
DB2	Lecture stable
Microsoft SQL Server	Lecture reproductible
Sybase Adaptive Server Enterprise	Lecture reproductible
Informix	Lecture reproductible

- Ajouts et suppressions verrouillés

Une transaction ne peut pas accéder à des lignes insérées ou supprimées depuis le début de la transaction.

Tableau 5. Types de base de données avec ajouts et suppressions verrouillés et niveaux d'isolement équivalents

Type de base de données	Niveau d'isolement équivalent
Oracle	Sans objet
DB2	Sans objet
Microsoft SQL Server	Sans objet
Sybase Adaptive Server Enterprise	Sans objet
Informix	Sans objet

- Transactions en séquence

Une série de transactions exécutées simultanément produit le même résultat que si elles étaient exécutées séquentiellement.

Tableau 6. Types de base de données sérialisable et niveaux d'isolement équivalents

Type de base de données	Niveau d'isolement équivalent
Oracle	Transactions en séquence
DB2	Lecture répétée
Microsoft SQL Server	Transactions en séquence
Sybase Adaptive Server Enterprise	Transactions en séquence
Informix	Sans objet

---

## Création de connexions de sources de données

Une source de données définit la connexion physique à une base de données. Une connexion de source de données spécifie les paramètres requis pour établir une connexion à une base de données, tels que l'emplacement de la base de données et le délai d'attente. Ces paramètres constituent une chaîne de connexion pour la source de données.

Vous pouvez créer des sources de données sur le portail ou dans Framework Manager. Etant donné qu'elles sont stockées sur le serveur, les sources de données apparaissent aux deux endroits, quel que soit l'emplacement où elles ont été créées. Il est possible d'éditer des connexions de sources de données uniquement dans le portail.

Si vous êtes administrateur, vous pouvez configurer toutes les sources de données requises avant que les modèles ne soient créés dans Framework Manager, de sorte que toutes les connexions soient disponibles dans l'assistant de métadonnées de Framework Manager.

Les sources de données sont stockées dans l'espace-noms Cognos et doivent avoir des noms uniques. Par exemple, vous ne pouvez pas utiliser le même nom pour une source de données et un groupe.

Vous pouvez inclure des informations d'authentification pour la base de données dans la connexion de source de données en créant un code d'accès. Les utilisateurs n'ont pas besoin de saisir des informations d'authentification chaque fois que la connexion est utilisée car les informations d'authentification sont chiffrées et stockées sur le serveur. Le code d'accès produit lors de la création d'une source de données est disponible pour le groupe Tous. Par la suite, vous pouvez changer les utilisateurs autorisés à utiliser le code d'accès ou créer d'autres codes d'accès.

Avant de créer des sources de données, vous devez disposer de droits d'accès en écriture sur le dossier dans lequel vous souhaitez enregistrer la source de données et sur l'espace-noms Cognos. Vous devez également disposer de droits d'exécution sur la fonction protégée **Connexions de sources de données**.

## Recommandation - Utilisation de chemins d'accès réseau pour les sources de données de type fichier

Si vous disposez d'une installation répartie avec plusieurs serveurs, il est conseillé d'utiliser des chemins d'accès réseau plutôt que des chemins d'accès locaux pour toutes les sources de données de type fichier. Les sources de données peuvent ainsi être consultées par les services qui en ont besoin, quel que soit le serveur qui requiert ces données.

Lorsque vous créez une connexion à une source de données de type fichier (par exemple, un PowerCube), vous devez saisir un chemin d'accès et un nom de fichier. Pour pointer vers le fichier, utilisez un chemin d'accès local, par exemple C:\cubes\Great Outdoors Company.mdc, ou un chemin d'accès réseau, par exemple \\servername\cubes\Great Outdoors Company.mdc.

Dans une installation répartie, les serveurs de rapports sont exécutés sur différents ordinateurs ; l'utilisation d'un chemin d'accès local suppose donc que le fichier et le chemin d'accès soient valides sur chaque ordinateur sur lequel le serveur de rapports est exécuté. Si vous utilisez un chemin d'accès réseau, chaque serveur de rapports doit pointer vers le même fichier sur le réseau, sans que le fichier soit nécessairement disponible en local. De plus, pour faire en sorte que le fichier soit toujours disponible, nous vous recommandons de stocker dans un répertoire partagé auquel il est possible d'accéder sur votre réseau.

Si vous avez installé des composants IBM Cognos Business Intelligence sur des serveurs UNIX, nous vous conseillons de localiser également la source de données de type fichier sur un serveur UNIX. Utilisez ensuite un chemin d'accès UNIX, tel que /nomserveur/cubes/Great Outdoors Company.mdc, pour accéder au fichier.

Si vous avez installé tous les composants sur un seul ordinateur, vous pouvez utiliser des chemins d'accès locaux, mais vous devez vous assurer que les services qui réclament les données disposent des droits d'accès appropriés sur les fichiers de données présents sur cet ordinateur.

Pour les installations réparties en environnement Microsoft Windows, nous vous conseillons d'utiliser des chemins d'accès UNC aux répertoires partagés pour toutes les sources de données de type fichier, par exemple des PowerCubes ou des fichiers XML.

## Connexions à des sources de données spécifiques

Pour créer une connexion à une source de données, vous devez tout d'abord consulter les informations pertinentes dans le document *IBM Cognos Administration and Security Guide*. Pour chacun des types de source de données suivants, vous trouverez des informations détaillées sur les paramètres de connexion et l'authentification requis.

- IBM DB2
- Cubes IBM Cognos
- Oracle Essbase
- IBM Infosphere Warehouse Cubing Services
- IBM Informix
- Microsoft Analysis Services
- Microsoft SQL Server
- Connexions ODBC

- Oracle
- SAP BW
- Sybase Adaptive Server Enterprise
- IBM Cognos TM1
- XML

Selon le type de la base de données à laquelle vous vous connectez, vous pouvez également être amené à vérifier les informations sur les niveaux d'isolement. Pour plus d'informations, voir «Niveaux d'isolement», à la page 40.

## Procédure de création d'une connexion de source de données

### Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Dans la fenêtre **Sélectionner la source de métadonnées**, cliquez sur **Sources de données**, puis sur **Suivant**.
4. Dans la fenêtre **Sélectionner la source de métadonnées**, cliquez sur **Nouveau**. Vous lancez ainsi l'assistant **Nouvelle source de données** qui est également disponible à partir de Cognos Connection.

A partir de l'**assistant Nouvelle source de données**, vous pouvez accéder au *Guide d'administration et de sécurité IBM Cognos* depuis le système d'aide en ligne.

5. Dans la page réservée au nom et à la description, saisissez un nom unique pour la connexion, ainsi qu'une description et une infobulle (facultatif), puis cliquez sur **Suivant**.
6. Dans la page de connexion, dans la liste déroulante **Type**, sélectionnez le type de source de données que vous souhaitez créer.  
Si votre source de données n'est pas répertoriée, sélectionnez **Autre type**.
7. Si nécessaire, spécifiez un niveau d'isolement pour votre source de données.
8. Saisissez tous les paramètres de la chaîne de connexion et indiquez d'autres paramètres, tels qu'un code d'accès ou un délai d'attente.  
Dans le *Guide d'administration et de sécurité IBM Cognos*, vous trouverez des informations détaillées sur les paramètres de connexion requis et sur l'authentification.

9. Cliquez sur **Test de la connexion**, puis sur **Tester** pour vérifier si les paramètres sont corrects.

Si le système vous y invite, saisissez un ID et un mot de passe utilisateur ou sélectionnez un code d'accès, puis cliquez sur **OK**. Si vous testez une connexion ODBC à un DSN utilisateur, vous devez être connecté en tant que créateur du DSN pour que le test aboutisse.

Dans la colonne **Statut**, vous pouvez vérifier si la connexion a été établie. Si elle n'a pas abouti, cliquez sur **Fermer**, revenez aux étapes précédentes et vérifiez vos paramètres de connexion.

### Résultats

Vous pouvez également tester la connexion à la source de données à partir du dossier **Sources de données** de l'**afficheur de projet**. Cliquez avec le bouton droit de la souris sur la source de données et cliquez sur **Tester**.

---

## Importation de métadonnées

Vous pouvez importer des métadonnées dans un nouveau projet ou dans un projet existant. L'importation de métadonnées est une opération qui peut être effectuée de nombreuses fois pour étendre le projet. Vous pouvez également exporter votre modèle dans un métamodèle CWM (Common Warehouse Metamodel).

IBM Cognos Framework Manager peut utiliser les métadonnées et les données provenant de sources de données externes pour créer un projet.

Pour importer des métadonnées, vous devez indiquer les sources souhaitées et leur emplacement. Vous ne pouvez importer qu'à partir d'une seule source de données à la fois. Si vous souhaitez importer à partir de plusieurs sources de données, vous devez effectuer plusieurs importations.

Vous pouvez importer des métadonnées à partir des sources suivantes :

- Bases de données relationnelles, telles que Oracle, IBM DB2 et Microsoft SQL Server
- Sources de données SAP BW
- Modèles IBM Cognos
- Modèles d'architecte et catalogues Impromptu
- Modèles IBM Cognos DecisionStream ou Data Manager
- Sources de métadonnées IBM
- Sources de métadonnées tierces
- XML en tant que source de données

Pour plus d'informations sur l'utilisation de connexions de sources de données, voir «Connexions de source de données», à la page 39.

### Noms d'objet en double

Lorsque vous importez des métadonnées, vous pouvez sélectionner la manière dont vous souhaitez que l'importation gère les noms d'objet en double. Vous pouvez ne pas importer l'objet ou importer et créer un nom unique. Lorsque vous choisissez d'importer tout sauf les objets en double, vous pouvez ajouter de nouveaux objets de base de données au projet sans les spécifier individuellement et sans avoir à exécuter de synchronisation. Pour importer des métadonnées avec des noms de table identiques, vous devez créer deux espaces-noms et importer chaque table dans un espace-noms distinct.

Lorsque vous importez des métadonnées SAP BW, IBM Cognos Framework Manager affecte un nom unique à chaque objet. Par conséquent, si vous renommez un objet dans le modèle et que vous le réimportez, Framework Manager détecte qu'il existe déjà. Pour réimporter un objet avec un autre nom unique, vous pouvez créer un nouvel espace-noms et réimporter l'objet dans ce dernier.

Vous pouvez également importer des objets ayant le même nom.

### Importation de métadonnées à partir de bases de données relationnelles

Lorsque vous importez des métadonnées à partir d'une base de données relationnelle, vous pouvez importer toutes les métadonnées ou sélectionner des types d'objet particuliers, par exemple, des tables, des colonnes, des vues, des



synonymes, des procédures stockées et des fonctions. Vous pouvez également importer des objets à partir d'une source relationnelle. Les procédures stockées système ne sont pas prises en charge. IBM Cognos Framework Manager ne prend en charge que les procédures stockées définies par l'utilisateur.

Le tableau suivant présente les objets de base de données qui sont mappés à des objets Framework Manager.

Objet de base de données	Objet Framework Manager
table	sujet de requête
colonne	élément de requête
vue	sujet de requête
synonyme	sujet de requête
procédure	sujet de requête
fonction	fonction de projet

Les ensembles nommés importés à partir de Microsoft SQL Server et de Microsoft Analysis Server sont stockés sous forme de calculs en lecture seule dans le modèle Framework Manager. Le calcul est associé à un indicateur qui l'identifie en tant qu'ensemble nommé et à une propriété qui contient le nom de la dimension.

IBM Cognos Business Intelligence (BI) ne prend pas en charge les packs Oracle qui sont essentiellement des groupes de procédures et de fonctions. Par conséquent, les synonymes pour les packs Oracle n'apparaissent pas dans Framework Manager. Les procédures stockées, les fonctions et les synonymes Oracle sont pris en charge dans Cognos BI et peuvent être importés à l'aide de l'assistant de métadonnées de Framework Manager dans la mesure où ils ne font pas référence à un pack Oracle.

## Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Dans la boîte de dialogue **Sélectionner la source de métadonnées**, choisissez le type de métadonnées que vous souhaitez importer.

L'option **Sources de données** fournit une liste des sources de données définies dans IBM Cognos Software.

- Sélectionnez la source de métadonnées relationnelles à importer.
- Sélectionnez une connexion de source de données et cliquez sur **Suivant**.  
Si la connexion de source de données souhaitée n'est pas répertoriée, vous devez d'abord la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.
- Cochez les cases correspondant aux objets que vous souhaitez importer.
- Spécifiez la manière dont l'importation doit traiter les noms d'objet en double. Indiquez si vous souhaitez importer et créer un nom unique, ou ne

pas importer. Si vous choisissez de créer un nom unique, l'objet importé s'affiche avec un numéro. Par exemple, vous verrez `SujetRequête1` dans votre projet.

- Si vous souhaitez importer des objets système, cochez la case **Afficher les objets système**, puis sélectionnez les objets système à importer.
- Indiquez les critères à utiliser pour créer des relations.
- Si vous souhaitez convertir tous les cardinalités en 1, désélectionnez la case à cocher **Fait de détection activée**. La suppression des cardinalités n désactive la détection des faits et empêche d'éviter automatiquement le double comptage. Pour plus d'informations, voir «Modélisation de relations de type 1-n en relations de type 1-1», à la page 370.

4. Cliquez sur **Importer**.

Les statistiques d'importation, ainsi qu'une liste des objets qui n'ont pas pu être importés et le nombre d'objets importés s'affichent.

5. Cliquez sur **Terminer**.

### Que faire ensuite

Une fois les métadonnées importées, vous devez vérifier les points suivants :

- Les relations et la cardinalité. Voir «Vérification des métadonnées importées», à la page 349.
- Les déterminants. Voir «Déterminants», à la page 84.
- La propriété **Utilisation** des éléments de requête. Voir «Formatage des éléments de requête», à la page 145.
- La propriété **Agrégat ordinaire** des éléments de requête. Voir «Formatage des éléments de requête», à la page 145.

## Importation de métadonnées à partir de modèles IBM Cognos

Vous pouvez importer des métadonnées à partir d'un modèle IBM Cognos existant.

**Remarque :** Si vous effectuez l'importation à partir d'un autre projet IBM Cognos Framework Manager, la syntaxe des expressions n'est pas ajustée pour chaque langue. Par exemple, vous créez un projet Framework Manager avec le français comme langue de conception et vous utilisez la syntaxe spécifique du français dans les calculs et les filtres. Vous créez ensuite un projet avec l'anglais comme langue de conception et vous importez le projet français dans le nouveau projet. Les expressions définies dans les calculs et les filtres ne sont pas valides. Vous devez modifier manuellement les expressions après l'importation des métadonnées.

### Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Cliquez sur **Modèle IBM Cognos, Suivant**.
4. Recherchez le modèle (fichier .cpf) souhaité, puis cliquez sur **Ouvrir, Suivant**.
5. Suivez les instructions de l'assistant **Importer** :
  - Cochez les cases correspondant aux objets que vous souhaitez importer.
  - Spécifiez la manière dont l'importation doit traiter les noms d'objet en double.

Indiquez si vous souhaitez importer et créer un nom unique, ou ne pas importer. Si vous choisissez de créer un nom unique, l'objet importé s'affiche avec un numéro. Par exemple, vous verrez `SujetRequête` et `SujetRequête1` dans votre projet.

6. Cliquez sur **Suivant, Terminer**.

## Importation de métadonnées à partir de modèles Architect ou de catalogues Impromptu

Pour importer des métadonnées à partir d'un modèle IBM Cognos Architect ou d'un catalogue Impromptu, vous devez d'abord les convertir en fichiers XML. En raison des différences entre IBM Cognos Series 7 et IBM Cognos BI, après l'importation des métadonnées migrées dans IBM Cognos Framework Manager, des tâches supplémentaires sont requises pour tester et affiner les métadonnées.

### Procédure

1. Vérifiez que vous avez exporté le modèle Architect ou le catalogue Impromptu.
2. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
3. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
4. Cliquez sur **IBM Cognos Architect (.xml)** ou **IBM Cognos Impromptu (.xml)**, puis cliquez sur **Suivant**.
5. Recherchez le fichier XML Architect ou Impromptu qui contient les métadonnées à importer.  
Un message dans la fenêtre **Aperçu XML** confirme que vous avez choisi un fichier XML valide.
6. Cliquez sur **Ouvrir**.
7. Sélectionnez l'espace-noms contenant vos informations de sécurité Series 7.
8. Cliquez sur **Importer**.  
La liste des objets créés s'affiche.
9. Si vous souhaitez vérifier les métadonnées importées, cochez la case **Vérifier après l'importation**.
10. Cliquez sur **Terminer**.

## Importation de métadonnées à partir d'IBM Cognos DecisionStream ou d'IBM Cognos Data Manager

Vous pouvez utiliser IBM Cognos Framework Manager pour importer des métadonnées à partir d'un fichier XML créé par IBM Cognos DecisionStream ou IBM Cognos Data Manager.

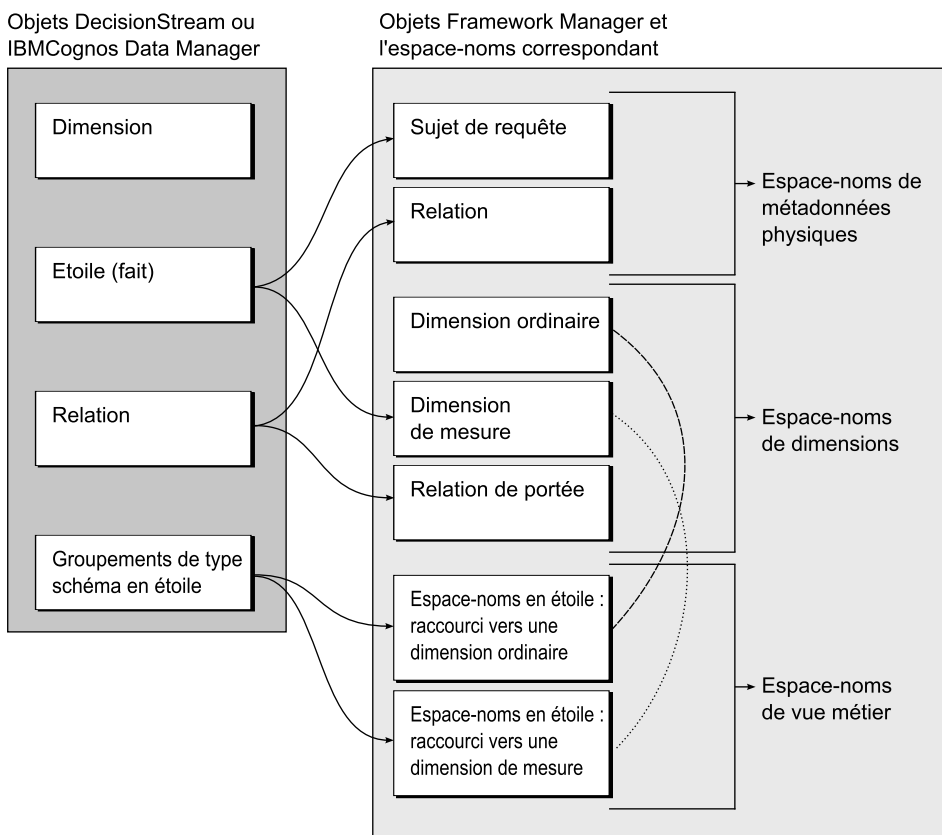
Vous pouvez importer les éléments suivants :

- Une couche physique résidant dans l'espace-noms **Métadonnées physiques**  
Cette couche contient des sujets de requête de source de données représentant les tables importées. La couche physique contient des sujets de requête et les relations physiques entre des sujets de requête. Ces relations physiques sont induites par les relations définies dans le fichier d'importation. Les tables importées deviennent des sujets de requête Framework Manager, et les clés secondaires deviennent des déterminants Framework Manager.
- Une couche dimensionnelle résidant dans l'espace-noms **Dimensions**

Cette couche contient des dimensions ordinaires, des dimensions de mesure et des relations de portée. Les dimensions ordinaires peuvent être conformes ou non conformes. Les objets de dimension de mesure correspondent aux étoiles importées. Les relations de portée sont induites par les relations définies dans le fichier d'importation.

- Une couche logique résidant dans l'espace-noms Vue métier  
 Cette couche contient des raccourcis vers les dimensions ordinaires et les dimensions de mesure contenues dans l'espace-noms Dimensions. Les raccourcis sont organisés en tant que groupements de schémas en étoile, qui sont des espaces-noms portant le même nom que les étoiles issues du fichier d'importation.

Le diagramme ci-après illustre le mappage des objets DecisionStream et Data Manager sur les objets Framework Manager et l'espace-noms correspondant.



## Faits

Une étoile est mappée sur un sujet de requête Framework Manager dans l'espace-noms Métadonnées physiques ou en tant que dimension de mesure dans l'espace-noms Dimensions. Le modèle inclut les attributs de fait suivants :

Nom d'attribut	Mappage Framework Manager
Nom de table	Nom du sujet de requête de base de données sur lequel est basée la requête de modèle représentant le fait

Nom d'attribut	Mappage Framework Manager
Nom abrégé	Description de sujet de requête
Nom métier	Propriété personnalisée
Description	Description de sujet de requête
Nom de colonne	Nom d'élément de requête
Type de colonne	Type des données d'élément de requête
Longueur de colonne	Taille d'élément de requête
Nom abrégé de colonne	Propriété personnalisée
Nom métier de colonne	Propriété personnalisée
Description de colonne	Description d'élément de requête
Type de colonne	Utilisation d'élément de requête
Clés de table	Déterminants résidant dans l'espace-noms Métadonnées physiques.

## Connexions

Une connexion est mappée sur une source de données Framework Manager.

**Remarque :** Une connexion de source de données n'est pas créée automatiquement dans IBM Cognos Connection. Vous devez créer manuellement la connexion dans IBM Cognos Connection, comme expliqué dans les étapes de la procédure d'importation. Pour plus d'informations, voir «Importation de métadonnées à partir d'IBM Cognos DecisionStream ou d'IBM Cognos Data Manager», à la page 49.

Le modèle inclut les attributs de source de données suivants :

Nom d'attribut	Mappage Framework Manager
Nom abrégé de connexion	Propriété personnalisée
Nom métier de connexion	Nom de la source de données
Description de connexion	Description de source de données
Connectivité	Propriété type.interface
Chaîne de connexion	Propriété personnalisée

## Génération de dimension

Une génération de dimension est mappée sur Framework Manager en tant qu'espace-noms de niveau supérieur.

## Hiéarchies

Une dimension contenant des hiérarchies, des niveaux, et des colonnes est mappée sur une dimension ordinaire Framework Manager contenant des hiérarchies, des niveaux et des éléments de requête.

## Etoiles conformes

Les étoiles conformes sont mappées sur un espace-noms Framework Manager résidant dans l'espace-noms Vue métier. Contient des raccourcis qui font référence aux dimensions.

Le modèle inclut les attributs d'étoile conforme suivants :

Nom d'attribut	Mappage Framework Manager
Nom abrégé d'étoile	Nom de l'espace-noms représentant l'étoile
Nom métier d'étoile	Nom de la dimension de mesure représentant le fait
Description d'étoile	Description de la dimension de mesure représentant le fait
Faits	Raccourcis vers une dimension de mesure
Dimensions	Raccourcis vers des dimensions ordinaires
Hiérarchies	Hiérarchies contenues dans la dimension ordinaire représentant les dimensions DecisionStream

## Propriétés de modèle

Le fichier d'exportation contient les propriétés de modèle ci-après.

Nom d'attribut	Mappage Framework Manager
Version de schéma	Pas de mappage
Version de catalogue	Propriété personnalisée
Nom abrégé de modèle	Nom de l'espace-noms représentant l'objet de modèle de niveau supérieur
Nom métier de modèle	Propriété personnalisée

Nom d'attribut	Mappage Framework Manager
Description de modèle	Description de l'espace-noms représentant l'objet de modèle de niveau supérieur

## Procédure d'importation à partir d'IBM Cognos DecisionStream ou IBM Cognos Data Manager

### Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Cliquez sur **IBM Cognos DecisionStream (.xml)** ou **IBM Cognos Data Manager (.xml)**, puis cliquez sur **Suivant**.
4. Recherchez le fichier XML qui contient les métadonnées à importer.  
Un message dans la fenêtre **Aperçu XML** confirme que vous avez choisi un fichier XML valide.
5. Cliquez sur **Ouvrir, Importer**.  
La liste des objets créés s'affiche.
6. Si vous souhaitez vérifier les métadonnées importées, cochez la case **Vérifier après l'importation**.
7. Cliquez sur **Terminer**.
8. Créez la connexion de source de données dans IBM Cognos Connection. Le nom que vous définissez dans IBM Cognos Connection doit être le même que celui affiché dans le dossier **Sources de données** dans Framework Manager.  
Pour des informations sur la façon de créer une connexion de source de données, voir «Création de connexions de sources de données», à la page 43.

## Importation à partir de sources de métadonnées IBM

Vous pouvez utiliser IBM Cognos Framework Manager pour importer des métadonnées à partir de sources de données IBM telles qu'IBM InfoSphere DataStage. Semblables à des sources de données tierces, les métadonnées sont extraites à l'aide de l'application MIMB (Meta Integration Model Bridge).

Pour plus d'informations, voir «Importation de métadonnées à partir de sources de métadonnées tiers», à la page 57.

### Options d'importation des sources de métadonnées IBM

Les options visibles varient en fonction de la source de données sélectionnée. Toutes les options ne s'appliquent pas à Framework Manager.

### Options de Framework Manager

Les options d'IBM Cognos Framework Manager disponibles sont toujours les mêmes, quel que soit le type de source de métadonnées que vous sélectionnez. Ce tableau présente les options utilisées pour créer des objets dans Framework Manager.

Options de Framework Manager	Description
Logical/Physical representation	<p>Indique comment les objets logiques et physiques sont représentés.</p> <p>La valeur <b>Integrated</b> représente les objets logiques et physiques comme un seul objet intégré.</p> <p>La valeur <b>Separated</b> représente les objets logiques et physiques comme deux objets connexes. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Separated (verbose)</b> représente les objets logiques et physiques comme deux objets distincts.</p>
Dimensional/Logical representation	<p>La valeur <b>Integrated</b> représente l'objet dimensionnel comme un seul objet intégré. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Separated</b> représente l'objet dimensionnel comme un objet dimensionnel et un objet logique.</p>
Diagram representation	<p>Indique la façon dont les diagrammes sont représentés dans le projet.</p> <p>La valeur <b>As Namespaces</b> indique que les diagrammes sont représentés comme des espaces-noms.</p> <p>La valeur <b>As Packages</b> indique que les diagrammes sont représentés comme des packs.</p> <p>La valeur <b>Both</b> indique que les diagrammes sont représentés à la fois comme des espaces-noms et comme des packs. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>None</b> indique que les diagrammes ne sont pas représentés dans le projet.</p>



Options de Framework Manager	Description
Namespace hierarchy	<p>Indique à quel endroit la hiérarchie de packs est conservée.</p> <p>La valeur <b>All</b> indique que la hiérarchie de packs est conservée dans tous les espaces-noms. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>None</b> indique que la hiérarchie de packs n'est pas conservée.</p> <p>La valeur <b>Dimensional level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms dimensionnel uniquement.</p> <p>La valeur <b>Logical level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms logique uniquement.</p> <p>La valeur <b>Physical level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms physique uniquement.</p>
Logical only classes	<p>Indique si les classes seulement logiques sont représentées.</p> <p>La valeur <b>Ignore</b> indique que les classes seulement logiques sont ignorées.</p> <p>La valeur <b>Show</b> indique que les classes seulement logiques sont affichées ainsi que tous leurs attributs. Il s'agit de la valeur par défaut.</p>
Logical only attributes	<p>Indique si les attributs seulement logiques sont représentés. Cette option n'affecte pas les attributs seulement logiques d'une classe seulement logique.</p> <p>La valeur <b>Drop</b> indique que les informations logiques d'une classe ne sont pas affichées si elles contiennent un attribut seulement logique.</p> <p>La valeur <b>Ignore</b> indique que les attributs seulement logiques sont ignorés.</p> <p>La valeur <b>Show</b> indique que les attributs seulement logiques sont affichés. Il s'agit de la valeur par défaut.</p>

Options de Framework Manager	Description
Unreferenced dimension attributes	<p>Indique la façon dont les attributs de dimension auxquels on ne fait pas référence sont représentés dans un projet. Un attribut de dimension auquel on ne fait pas référence est un attribut de dimension qui ne fait pas partie d'un niveau et qui n'est référencé par aucun autre attribut de dimension.</p> <p>La valeur <b>Ignore All</b> indique tous les attributs de dimension auxquels on ne fait pas référence sont ignorés.</p> <p>La valeur <b>Ignore Join</b> indique que seuls les attributs de dimension auxquels on ne fait pas référence qui font partie d'une jointure sont ignorés.</p> <p>La valeur <b>Show All</b> indique tous les attributs de dimension auxquels on ne fait pas référence sont affichés. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Show Join</b> indique que seuls les attributs de dimension auxquels on ne fait pas référence qui font partie d'une jointure sont affichés.</p>
Dimensions without dimensional information	<p>Indique comment représenter les dimensions qui ne contiennent aucune information dimensionnelle.</p> <p>La valeur <b>Dimension</b> crée une dimension et une hiérarchie et un niveau par défaut. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Requête de modèle</b> crée une requête de modèle.</p>
Create cubes	<p>Indique si l'importation peut créer des cubes. Un cube est représenté sous la forme d'un espace-noms contenant toutes les informations nécessaires pour créer le cube.</p>
Populate screentip	<p>Indique si l'importation utilise la zone de description en tant qu'infobulle.</p> <p>La valeur <b>True</b> permet au contenu de la zone de description d'être affiché sous la forme d'une infobulle dans IBM Cognos Report Studio, Query Studio ou IBM Cognos Workspace Advanced.</p>
Name of the namespace containing the dimensional information	<p>Indique l'espace-noms qui contient les informations dimensionnelles. La valeur par défaut est "Modèle dimensionnel".</p>

Options de Framework Manager	Description
Name of the namespace containing the logical information	Indique l'espace-noms qui contient les informations logiques. La valeur par défaut est "Modèle logique".
Name of the namespace containing the physical information	Indique l'espace-noms qui contient les informations physiques. La valeur par défaut est "Modèle physique".
Name of the namespace containing the subject areas	Indique l'espace-noms qui contient les zones de sujet. La valeur par défaut est "Zone de sujet".
Vérification de l'uniformité	Indique le niveau de vérification de l'uniformité.  La valeur <b>De base</b> correspond au niveau de vérification de l'uniformité standard.  La valeur <b>Extensive</b> permet d'effectuer une validation plus approfondie du modèle.  La valeur <b>None</b> indique qu'aucune validation n'est effectuée.

## Importation de métadonnées à partir de sources de métadonnées tiers

Vous pouvez utiliser IBM Cognos Framework Manager pour importer des métadonnées à partir d'autres sources, y compris des métadonnées relationnelles. Les métadonnées sont importées à l'aide d'un pont de métadonnées.

Vous pouvez importer à la fois des métadonnées relationnelles et d'autres métadonnées dans le même modèle. Vous pouvez commencer avec un nouveau modèle Framework Manager et importer les autres métadonnées avant les métadonnées relationnelles. Cela permet d'éviter tout conflit si vous importez des objets portant le même nom.

Lorsque vous importez d'autres métadonnées, des sources de données sont créées en fonction des informations fournies par le biais de l'assistant d'importation. Si au moins un objet physique présent dans l'autre source fait référence à un schéma de base de données et/ou à un catalogue, une source de données Framework Manager est créée et ses propriétés de catalogue ou de schéma prennent les noms du catalogue ou du schéma spécifiés dans les métadonnées. Une source de données générique est créée pour les objets physiques qui ne font pas référence à un catalogue ni à un schéma.

Si vous souhaitez accéder à des métadonnées à partir d'une autre source de données, vous devez effectuer une importation de modèle physique.

Les sources de données ne contiennent pas toutes des métadonnées appropriées pour la génération de rapports Business Intelligence et les concepts ne sont pas tous adaptés à Framework Manager. L'importation des métadonnées est adaptée à Framework Manager et seules les métadonnées compatibles seront importées.

Tous les ponts de métadonnées offrent une couche physique qui fournit la base pour une modélisation supplémentaire. La richesse du modèle Framework Manager obtenu est directement liée à la richesse de la source de métadonnées.

## Bases de données multiples

Les autres sources de métadonnées peuvent être basées sur des bases de données multiples. La méthode la plus efficace pour importer ces sources de données multiples dans Framework Manager consiste à effectuer des importations multiples. Pour chaque importation, vous sélectionnez les éléments qui correspondent à cette source de données spécifique.

Par exemple, la première fois que vous effectuez une importation à partir d'une autre source de métadonnées, vous sélectionnez `datasource1` et tous les éléments qui correspondent à cette source de données. La fois suivante, vous sélectionnez `datasource2` et les éléments qui correspondent à cette source de données. Vous poursuivez l'opération d'importation tant que vous n'avez pas importé toutes les sources de données et leurs éléments correspondants.

## Options d'importation spécifiques à un élément tiers

Les métadonnées sont extraites des sources de données provenant d'éléments tiers par l'application MIMB (Meta Integration Model Bridge). Toutes les options ne s'appliquent pas à Framework Manager. Pour plus d'informations sur les outils et les mappages d'objet pris en charge, voir le site Web relatif à Meta Integration.

Les options visibles lors de l'importation varient en fonction de la source de données sélectionnée.

## Options de Framework Manager

Les options d'IBM Cognos Framework Manager disponibles sont toujours les mêmes, quel que soit le type de source de métadonnées que vous sélectionnez. Ce tableau présente les options utilisées pour créer des objets dans Framework Manager.

Options de Framework Manager	Description
Logical/Physical representation	<p>Indique comment les objets logiques et physiques sont représentés.</p> <p>La valeur <b>Integrated</b> représente les objets logiques et physiques comme un seul objet intégré.</p> <p>La valeur <b>Separated</b> représente les objets logiques et physiques comme deux objets connexes. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Separated (verbose)</b> représente les objets logiques et physiques comme deux objets distincts.</p>

Options de Framework Manager	Description
Dimensional/Logical representation	<p>La valeur <b>Integrated</b> représente l'objet dimensionnel comme un seul objet intégré. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Separated</b> représente l'objet dimensionnel comme un objet dimensionnel et un objet logique.</p>
Diagram representation	<p>Indique la façon dont les diagrammes sont représentés dans le projet.</p> <p>La valeur <b>As Namespaces</b> indique que les diagrammes sont représentés comme des espaces-noms.</p> <p>La valeur <b>As Packages</b> indique que les diagrammes sont représentés comme des packs.</p> <p>La valeur <b>Both</b> indique que les diagrammes sont représentés à la fois comme des espaces-noms et comme des packs. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>None</b> indique que les diagrammes ne sont pas représentés dans le projet.</p>
Namespace hierarchy	<p>Indique à quel endroit la hiérarchie de packs est conservée.</p> <p>La valeur <b>All</b> indique que la hiérarchie de packs est conservée dans tous les espaces-noms. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>None</b> indique que la hiérarchie de packs n'est pas conservée.</p> <p>La valeur <b>Dimensional level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms dimensionnel uniquement.</p> <p>La valeur <b>Logical level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms logique uniquement.</p> <p>La valeur <b>Physical level only</b> indique que la hiérarchie de packs est conservée dans l'espace-noms physique uniquement.</p>

Options de Framework Manager	Description
Logical only classes	<p>Indique si les classes seulement logiques sont représentées.</p> <p>La valeur <b>Ignore</b> indique que les classes seulement logiques sont ignorées.</p> <p>La valeur <b>Show</b> indique que les classes seulement logiques sont affichées ainsi que tous leurs attributs. Il s'agit de la valeur par défaut.</p>
Logical only attributes	<p>Indique si les attributs seulement logiques sont représentés. Cette option n'affecte pas les attributs seulement logiques d'une classe seulement logique.</p> <p>La valeur <b>Drop</b> indique que les informations logiques d'une classe ne sont pas affichées si elles contiennent un attribut seulement logique.</p> <p>La valeur <b>Ignore</b> indique que les attributs seulement logiques sont ignorés.</p> <p>La valeur <b>Show</b> indique que les attributs seulement logiques sont affichés. Il s'agit de la valeur par défaut.</p>
Unreferenced dimension attributes	<p>Indique la façon dont les attributs de dimension auxquels on ne fait pas référence sont représentés dans un projet. Un attribut de dimension auquel on ne fait pas référence est un attribut de dimension qui ne fait pas partie d'un niveau et qui n'est référencé par aucun autre attribut de dimension.</p> <p>La valeur <b>Ignore All</b> indique tous les attributs de dimension auxquels on ne fait pas référence sont ignorés.</p> <p>La valeur <b>Ignore Join</b> indique que seuls les attributs de dimension auxquels on ne fait pas référence qui font partie d'une jointure sont ignorés.</p> <p>La valeur <b>Show All</b> indique tous les attributs de dimension auxquels on ne fait pas référence sont affichés. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Show Join</b> indique que seuls les attributs de dimension auxquels on ne fait pas référence qui font partie d'une jointure sont affichés.</p>

Options de Framework Manager	Description
Dimensions without dimensional information	<p>Indique comment représenter les dimensions qui ne contiennent aucune information dimensionnelle.</p> <p>La valeur <b>Dimension</b> crée une dimension et une hiérarchie et un niveau par défaut. Il s'agit de la valeur par défaut.</p> <p>La valeur <b>Requête de modèle</b> crée une requête de modèle.</p>
Create cubes	<p>Indique si l'importation peut créer des cubes. Un cube est représenté sous la forme d'un espace-noms contenant toutes les informations nécessaires pour créer le cube.</p>
Populate screentip	<p>Indique si l'importation utilise la zone de description en tant qu'infobulle.</p> <p>La valeur <b>True</b> permet au contenu de la zone de description d'être affiché sous la forme d'une infobulle dans IBM Cognos Report Studio, Query Studio ou IBM Cognos Workspace Advanced.</p>
Name of the namespace containing the dimensional information	<p>Indique l'espace-noms qui contient les informations dimensionnelles. La valeur par défaut est "Modèle dimensionnel".</p>
Name of the namespace containing the logical information	<p>Indique l'espace-noms qui contient les informations logiques. La valeur par défaut est "Modèle logique".</p>
Name of the namespace containing the physical information	<p>Indique l'espace-noms qui contient les informations physiques. La valeur par défaut est "Modèle physique".</p>
Name of the namespace containing the subject areas	<p>Indique l'espace-noms qui contient les zones de sujet. La valeur par défaut est "Zone de sujet".</p>
Vérification de l'uniformité	<p>Indique le niveau de vérification de l'uniformité.</p> <p>La valeur <b>De base</b> correspond au niveau de vérification de l'uniformité standard.</p> <p>La valeur <b>Extensive</b> permet d'effectuer une validation plus approfondie du modèle.</p> <p>La valeur <b>None</b> indique qu'aucune validation n'est effectuée.</p>

## Procédure d'importation à partir de sources de métadonnées tiers

### Avant de commencer

Pour pouvoir importer des métadonnées, une connexion doit être établie avec la source de données.

### Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Cliquez sur **Sources de métadonnées tiers, Suivant**.
4. Cliquez sur le type de métadonnées à importer.
5. Dans la boîte de dialogue **Options d'importation spécifiques à un élément tiers**, utilisez l'option **Fichier** pour identifier le fichier qui doit contenir les métadonnées à importer.

Cliquez sur les autres options souhaitées. Les options visibles varient en fonction de la source de données sélectionnée.

**Remarque :** Utilisez les options par défaut. Ces options permettent d'optimiser l'importation des métadonnées. Si vous modifiez les options, des résultats inattendus risquent de se produire. Pour rétablir les options par défaut, cliquez sur **Utiliser les valeurs par défaut**.

6. Cliquez sur **Suivant**.
7. Dans la boîte de dialogue **Options d'importation spécifiques à Framework Manager**, cliquez sur les options souhaitées, puis cliquez sur **Suivant**.
8. Suivez les instructions de l'**assistant de métadonnées** :
  - Sélectionnez une connexion de source de données et cliquez sur **Suivant**.  
Si la connexion de source de données souhaitée n'est pas répertoriée, vous devez d'abord la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.
  - Cochez les cases correspondant aux objets que vous souhaitez importer.
  - Spécifiez la manière dont l'importation doit traiter les noms d'objet en double. Indiquez si vous souhaitez importer et créer un nom unique, ou ne pas importer. Si vous choisissez de créer un nom unique, l'objet importé s'affiche avec un numéro. Par exemple, vous verrez **SujetRequête** et **SujetRequête1** dans votre projet.
  - Spécifiez les critères à utiliser pour créer des relations et cliquez sur **Importer**.  
Pour plus d'informations, voir «Relations», à la page 68.

Les statistiques d'importation, ainsi qu'une liste des objets qui n'ont pas pu être importés et le nombre d'objets importés s'affichent.
9. Cliquez sur **Terminer**.

### Modélisation après l'importation :

Les métadonnées source ont une structure logique qui est compatible avec IBM Cognos BI. Lors de l'importation, cette structure est conservée pour une meilleure fidélité au modèle source.

Après l'importation, seules les métadonnées physiques sont disponibles.



### Procédure

1. Analysez et vérifiez le modèle. Voir «Analyse de modèles», à la page 190.
2. Définissez les déterminants.
3. Définissez la propriété **Utilisation**.  
Certains ponts nécessitent que des faits soient définis manuellement car ils ne sont pas disponibles dans la source de métadonnées.
4. Vérifiez les relations et la cardinalité.
5. Résolvez les relations ambiguës, comme les relations valides multiples, les relations réflexives et les relations récursives.

## Traitement des incidents liés aux métadonnées issues d'autres sources

Cette section décrit les problèmes que vous pouvez rencontrer lors de l'utilisation de métadonnées que vous avez importées dans IBM Cognos Framework Manager.

### Des relations impliquant des vues de table ne sont pas importées à partir d'un fichier Oracle Designer

Des relations de clé primaire et externe impliquant au moins une table de vue ne sont pas importées à partir d'un fichier Oracle Designer dans Framework Manager.

Créez manuellement les relations de clé primaire et externe.

### Impossible de tester un sujet de requête à partir d'une autre source

Vous ne pouvez pas tester un sujet de requête que vous avez importé à partir d'une autre source de données.

Par exemple, vous testez un sujet de requête et vous obtenez ce message d'erreur :

QE-DEF-0177 Une erreur s'est produite lors de l'exécution de l'opération 'sqlPrepareWithOptions' statut = '-201'.

UDA-SQL-0196 La table ou la vue "GOSALES1.COGNOS.COM.GOSALES.CONVERSION\_RATE" est introuvable dans le dictionnaire.

Pour résoudre ce problème, assurez-vous que les conditions suivantes sont satisfaites :

- L'objet de source de données Framework Manager créé par l'importation et référencé par le sujet de requête dispose d'une source de données correspondante portant le même nom dans le magasin de contenu IBM Cognos Business Intelligence.
- La source de données correspondante du magasin de contenu est valide et les informations de connexion sont correctes.
- Les propriétés facultatives de catalogue ou de schéma de l'objet de source de données créé par l'importation ainsi que les références au sujet de requête sont correctes. Pour les bases de données acceptant les identificateurs sensibles à la casse, vérifiez également que la casse est correcte.
- L'objet de base de données (table ou vue) représenté par le sujet de requête existe dans la base de données avec le même nom et est accessible au sein de la connexion en cours.
- Le **type de requête** est valide. Il peut exister un **type de requête** non valide si vous importez un sujet de requête à partir de SAP BW et que vous modifiez la

valeur de la propriété **Type de requête** pour la source de données. Par défaut, la valeur de **Type de requête** est **Multidimensionnel**. Ne la modifiez pas.

### **Certaines expressions importées à partir d'autres sources de métadonnées ne sont pas valides**

La prise en charge de l'analyse syntaxique d'expression a été améliorée dans le MIMB d'IBM Cognos BI. Les fonctions qui sont équivalentes entre d'autres sources de métadonnées et IBM Cognos BI font l'objet d'une analyse syntaxique.

Les exceptions sont les suivantes :

- Fonctions sans mappage entre IBM Cognos BI et les autres sources de métadonnées
- Expressions qui utilisent une syntaxe spécialisée

#### **Solution :**

Si vous disposez de fonctions sans mappage entre IBM Cognos BI et les autres sources de métadonnées, après avoir importé des expressions à partir d'autre sources de métadonnées, modifiez ces expressions manuellement dans Framework Manager pour qu'elles soient conformes à la syntaxe Framework Manager équivalente.

Si vous disposez d'expressions qui utilisent une syntaxe spécifique, procédez comme suit :

#### **Procédure**

1. Dans Framework Manager, identifiez tous les éléments de requête importés qui représentent des calculs intégrés à l'aide de références telles que \$\$1, \$\$2, etc.
2. Dans la sous-fenêtre **Propriétés**, recherchez les informations fournies dans la propriété **patternMapping** pour chaque élément de requête.  
La propriété **patternMapping** indique le mappage entre les paramètres du calcul importé et les références d'objet réel du modèle d'origine.
3. Cliquez deux fois sur un sujet de requête altéré.
4. Cliquez deux fois sur le calcul intégré correspondant à l'élément de requête identifié à l'étape 1.
5. Si un nom par défaut ("Calculation...") a été attribué au calcul, remplacez-le par le nom d'élément de requête réel.
6. Remplacez les paramètres importés par les références d'objet réel Framework Manager que ces paramètres représentent.
7. Répétez cette étape pour chaque sujet de requête altéré.

## **Importation de métadonnées en utilisant un fichier XML comme source de données**

Vous pouvez importer un fichier XML sous la forme d'une source de données tabulaires dans IBM Cognos Framework Manager. Vous pouvez l'importer en local ou à partir d'un site distant via une adresse URL valide. Dans Framework Manager, le fichier XML est utilisé pour modéliser des métadonnées et créer un pack.

Le fichier XML est validé et analysé lors de l'exécution lorsque la requête est traitée par Report Studio, Query Studio ou IBM Cognos Workspace Advanced. Si vous ajoutez l'option **VALIDATE=ON** à la chaîne de connexion, Framework Manager valide partiellement le fichier XML dans la balise `<columnList>` qui décrit les

métadonnées. Pour plus d'informations sur les types de données pris en charge, voir Annexe H, «Types de données XML», à la page 671.

Vous devez utiliser le schéma `xmldata.xsd` pour valider le fichier XML. Le schéma est situé dans le dossier `\c10\bin`. Il n'est pas nécessaire de spécifier l'emplacement du schéma dans le fichier XML lui-même.

Pour utiliser un fichier XML comme source de données, vérifiez que:

- Vous n'utilisez pas SQL natif pour accéder aux données dans un fichier XML
- Vous n'accédez pas à des objets BLOB (Binary Large Object)
- Vous utilisez uniquement des appels de métadonnées `sqlColumns()` et `sqlTables()`.

Les autres appels renvoient une erreur de fonction non prise en charge

- Le fichier XML a une structure correcte et est valide

## Avant de commencer

Pour pouvoir importer des métadonnées, une connexion doit être établie avec la source de données. Si la source de données XML se trouve sur un autre ordinateur, vous devez utiliser un compte qui dispose des droits nécessaires pour accéder à la source de données.

Une fois que vous avez établi une connexion à une source de données XML, la source de données s'affiche dans la liste des sources de données.

Pour utiliser XML comme source de données, vous devez comprendre le langage XML, les schémas et les autres technologies liées à XML.

## Procédure

1. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation.
2. Cliquez sur **Actions, Exécuter l'assistant de métadonnées**.
3. Cliquez sur la source de données XML à importer, puis sur **Suivant**.

## Importation d'objets portant le même nom

Les espaces-noms sont des conteneurs similaires à des dossiers. Les objets d'un projet IBM Cognos Framework Manager doivent être identifiables de manière unique. Si deux objets portent le même nom, ils doivent résider dans deux espaces-noms distincts.

Par exemple, vous disposez d'une base de données contenant des données financières. Un ensemble de tables représente les prévisions et un autre, les données réelles. Les prévisions et les données réelles comprennent des tables nommées Comptes fournisseurs et Comptes clients. Pour importer ces tables dans Framework Manager et utiliser les mêmes noms de table dans le projet, vous devez créer deux espaces-noms. Vous pouvez nommer un espace-noms Prévisions et l'autre, Réelles.

## Procédure

1. Cliquez sur l'espace-noms de modèle ou l'espace-noms racine, puis cliquez sur **Actions, Créer, Espace-noms**.
2. Cliquez avec le bouton droit de la souris sur l'espace-noms, cliquez sur **Renommer** et entrez un nom descriptif.



---

## Chapitre 5. Modélisation des métadonnées relationnelles

**Remarque :** Les informations sur les métadonnées SAP BW figurent dans des rubriques distinctes.

Après avoir importé des métadonnées, vous devez vous assurer qu'elles sont configurées selon les besoins de vos utilisateurs en termes de rapports et fournir les informations supplémentaires requises. Les améliorations que vous apportez à IBM Cognos Framework Manager n'ont pas d'incidence sur la source de données d'origine.

**Astuce :** Pour vérifier que le modèle répond aux besoins en termes de rapports, vous pouvez sélectionner des objets à afficher dans un rapport et les tester. Les résultats du test montrent le rapport que vos utilisateurs verront, ainsi que le code SQL et les messages issus du logiciel IBM Cognos, le cas échéant. Vous pouvez également publier un pack à tout moment, puis l'utiliser pour créer des rapports.

Vous pouvez vérifier le projet quand bon vous semble pour vous assurer que les références entre les objets qu'il contient sont valides. Voir «Vérification des modèles ou des packs», à la page 255.

Créez au moins deux vues : la vue d'importation et la vue métier. L'utilisation de deux vues ou plus facilite le remappage d'éléments à une nouvelle source de données.

---

### Configuration de la vue d'importation

La vue d'importation vue les métadonnées qui ont été importées à partir de la source de données.

Pour vous assurer que les métadonnées sont configurées correctement dans la vue d'importation, procédez comme suit :

#### Procédure

1. Vérifiez que les relations correspondent aux besoins en matière de génération de rapports.
2. Optimisez et personnalisez les données extraites par les sujets de requête.
3. Optimisez et personnalisez les données extraites par les dimensions. Vous pouvez stocker les dimensions dans une vue dimensionnelle distincte.
4. Traitez la prise en charge des métadonnées multilingues.
5. Contrôlez la manière dont les données sont utilisées et formatées en vérifiant les propriétés d'élément de requête.

---

### Configuration de la vue métier

La vue métier fournit une couche pour les informations de la source de données pour les utilisateurs puissent générer des rapports plus facilement.

Pour étendre les métadonnées de la vue métier, procédez comme suit :

## Procédure

1. Ajoutez des règles métier, telles que des calculs et des filtres, pour définir les informations que les utilisateurs peuvent extraire.
2. Organisez le modèle en créant pour chaque groupe d'utilisateurs des vues distinctes reflétant les concepts métier que les utilisateurs connaissent bien.
3. Créez un modèle durable capable de résister aux modifications ultérieures de noms d'élément de requête sans impact sur les rapports existants et les auteurs de rapports.

## Résultats

Vous pouvez analyser les métadonnées pour vous assurer que le modèle respecte les bonnes pratiques de modélisation actuelles suivantes à l'aide de **Model Advisor**.

---

## Relations

Une relation explique comment créer une requête relationnelle pour plusieurs objets dans le modèle. Sans relation, ces objets sont des ensembles de données isolés.

Les relations fonctionnent dans les deux directions. Vous devez souvent examiner les deux directions pour comprendre complètement la relation.

Les différents types de relations sont les suivants

- Un à un  
Des relations un à un ont lieu lorsqu'une instance de données d'un sujet de requête est liée à exactement une instance d'un autre sujet de requête. Par exemple, chaque étudiant est associé à un numéro d'étudiant.
- Un à plusieurs ou zéro à plusieurs  
Des relations un à plusieurs ou zéro à plusieurs ont lieu lorsqu'une instance de données d'un sujet de requête est liée à plusieurs instances d'un autre sujet de requête. Par exemple, professeur a plusieurs étudiants.
- Multivoque  
Des relations multivoques ont lieu lorsque plusieurs instances de données d'un sujet de requête sont liées à plusieurs instances d'un autre sujet de requête. Par exemple, plusieurs professeurs ont plusieurs étudiants.

Lors de l'importation de métadonnées, IBM Cognos Framework Manager peut créer des relations entre les objets du modèle en fonction des clés primaires et externes de la source de données. Vous pouvez créer ou supprimer des relations dans le modèle pour que ce dernier représente au mieux la structure logique de votre entreprise.

Après avoir importé les métadonnées, vérifiez que les relations dont vous avez besoin existent dans le projet et que la cardinalité est correctement définie. La source de données peut avoir été conçue sans utiliser l'intégrité référentielle. Souvent, de nombreuses contraintes de clé primaire et unique ne sont pas spécifiées. Sans ces contraintes, Framework Manager ne peut pas générer les relations nécessaires entre les tables de faits et les tables de dimension.

Framework Manager stocke les relations dans le plus proche parent commun des objets qui participent à la relation. Le parent peut être un dossier ou un espace-noms. Si vous déplacez l'un des objets participants en dehors du parent

commun, la relation se déplace vers l'espace-noms suivant qui est commun aux deux extrémités de la relation. Si vous déplacez une relation dans un autre dossier ou espace-noms, les objets participants sont également déplacés vers le même dossier ou espace-noms.

**Astuce :** Utilisez l'onglet **Rechercher** (sous-fenêtre **Outils**) pour rechercher un objet de la classe Relation dont le nom correspond à un modèle spécifié. Par exemple, si vous recherchez une relation dont le nom contient En-tête de commande, Framework Manager trouve toutes les relations dont une extrémité comporte En-tête de commande. Si vous avez renommé une relation, une recherche de ce type peut ne pas aboutir.

## Cardinalité

Des relations existent entre deux sujets de requête. La cardinalité d'une relation est le nombre de lignes liées à chacun des deux sujets de requête. Les lignes sont liées par l'expression de la relation ; cette expression fait généralement référence aux clés primaires et étrangères des tables sous-jacentes.

Le logiciel IBM Cognos utilise la cardinalité d'une relation pour :

- éviter le double comptage des données de fait ;
- rendre en charge les jointures de boucles qui sont fréquentes dans les modèles à schéma en étoile ;
- optimiser l'accès au système de la source de données sous-jacente ;
- identifier les sujets de requête qui se comportent comme des faits ou dimensions.

Une requête qui utilise plusieurs faits de différentes tables sous-jacentes est divisée en requêtes distinctes pour chaque table de faits sous-jacente. Chaque requête de fait individuelle fait référence à sa table de faits correspondante ainsi qu'aux tables dimensionnelles liées à cette table de faits. Une autre requête est utilisée pour fusionner ces requêtes individuelles en ensemble unique de résultats. La requête de cette dernière opération est généralement appelée requête liée. Vous savez qu'il s'agit d'une requête liée lorsque vous pouvez voir `coalesce` ainsi qu'une jointure externe intégrale.

La requête liée permet également au logiciel IBM Cognos de mettre correctement des données en rapport à divers niveaux de granularité. Voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

Vous devez vous assurer que toutes les relations et la cardinalité correspondent aux besoins des utilisateurs.

Pour plus d'informations, voir «Cardinalité des requêtes générées», à la page 328 et «Cardinalité dans le contexte d'une requête», à la page 329.

### Détection de la cardinalité à partir d'une source de données

Lorsque vous effectuez une importation à partir d'une source de données relationnelles, la cardinalité est détectée à partir d'un ensemble de règles que vous définissez.

Les options disponibles sont les suivantes :

- Utiliser les clés principales et secondaires

- Utiliser les noms d'élément de requête correspondants qui représentent des colonnes indexées de manière unique
- Utiliser les noms des éléments de requête correspondants

La situation la plus fréquente consiste à utiliser les clés primaires et externes ainsi que les éléments de requête correspondants qui représentent des colonnes indexées de manière unique. Les informations sont utilisées pour définir certaines propriétés des éléments de requête ainsi que pour générer des relations.

Pour afficher les informations d'index et de clé qui ont été importées, cliquez avec le bouton droit de la souris sur un sujet de requête, puis cliquez sur **Editer la définition**. Pour un sujet de requête, vous pouvez modifier les informations dans l'onglet **Déterminants**.

Les relations facultatives, jointures externes complètes et relations multivoques peuvent être importées à partir de votre source de données. Elles seront exécutées comme des requêtes par IBM Cognos Framework Manager.

**Remarque :** toutes les dimensions ordinaires commencent comme des sujets de requête. Si vous convertissez un sujet de requête en une dimension ordinaire, vous constatez que les informations de déterminant du sujet de requête sont optimisées comme point de départ de la définition des niveaux d'une hiérarchie unique. Vérifiez les niveaux et les clés créés dans la hiérarchie de la dimension.

## Notation

Par défaut, Framework Manager utilise la notation Merise. La notation Merise marque chaque extrémité de la relation avec la cardinalité minimale et maximale de cette extrémité. Vous pouvez également utiliser la notation Crowsfeet qui fournit une représentation illustrée de la relation.

Pour plus d'informations sur la procédure de modification de la notation, voir «Modification des paramètres de diagrammes», à la page 16.

Lorsque vous interprétez la cardinalité, vous devez tenir compte de la notation qui s'affiche à chaque extrémité de la relation.

Les libellés d'extrémité possibles sont les suivants :

- 0..1 (aucune ou une correspondance)
- 1..1 (une seule correspondance)
- 0..n (aucune ou plusieurs correspondances)
- 1..n (une ou plusieurs correspondances)

La première partie de la notation indique le type de jointure pour cette relation :

- Une jointure interne (1)  
Une jointure interne affiche toutes les lignes correspondantes des deux objets.
- Une jointure externe (0)  
Une jointure externe affiche tous les éléments des deux objets, y compris les éléments qui ne correspondent pas. Une jointure externe peut être qualifiée comme étant complète, de gauche ou de droite. Les jointures externes sur l'élément de droite et sur l'élément de gauche prennent tous les éléments du côté droit ou gauche de la relation, respectivement, et uniquement ce qui correspond de l'autre côté.



Vos utilisateurs visualisent un rapport différent selon que vous utilisez une jointure interne ou externe. Par exemple, vos utilisateurs souhaitent obtenir un rapport qui répertorie les vendeurs et les commandes. Si vous utilisez une jointure externe pour connecter les vendeurs et les commandes, le rapport affiche tous les vendeurs, qu'ils soient associés ou non à des commandes. Si vous utilisez une jointure interne, le rapport affiche seulement les vendeurs qui ont passé des commandes.

Les données présentes dans un objet peuvent ne correspondre à aucun élément de l'autre objet. Toutefois, si la relation possède une cardinalité minimale de 1, une jointure interne est toujours utilisée et ces enregistrements ne sont pas pris en compte. Inversement, si tous les éléments correspondent mais que la relation du modèle présente une cardinalité minimale de 0, une jointure externe est toujours utilisée, même si les résultats sont les mêmes que ceux obtenus avec une jointure interne. Par exemple, la table sous-jacente d'un objet contient une clé externe obligatoire (ne pouvant avoir la valeur nulle) pour l'autre objet. Assurez-vous que les données et les cardinalités correspondent.

La seconde partie de la notation définit la relation des éléments de requête entre les objets.

### Cardinalité des requêtes générées

Le logiciel IBM Cognos prend en charge la cardinalité minimale/maximale et la cardinalité facultative.

Dans 0:1, 0 est la cardinalité minimale et 1 est la cardinalité maximale.

Dans 1:n, 1 est la cardinalité minimale et n est la cardinalité maximale.

Une relation dont la cardinalité est définie comme 1:1 à 1:n est généralement considérée en tant que 1 à n lors de l'analyse des cardinalités maximales.

Une cardinalité minimale de 0 signifie que la relation est facultative. Vous pouvez définir une cardinalité minimale de 0 lorsque vous souhaitez que la requête conserve les informations de l'autre côté de la relation en l'absence de correspondance. Par exemple, une relation entre le client et les ventes réelles peut être définie comme 1:1 à 0:n. Cela indique que les rapports vont afficher les informations relatives au client demandé même s'il n'y a aucune donnée sur les ventes.

Par conséquent, une relation 1 à n peut également être indiquée sous la forme suivante :

- 0:1 à 0:n
- 0:1 à 1:n
- 1:1 à 0:n
- 1:1 à 1:n

Utilisez l'instruction **Incidence de la relation** de la boîte de dialogue **Définition de relation** pour mieux comprendre la cardinalité. Par exemple, Personnel de vente (1:1) est joint à Commandes (0:n).

Relationship impact:	Each Order has one and only one Sales Staff. Each Sales Staff has zero or more Order (outer join).
----------------------	---

Il est important de vérifier que la cardinalité est correctement prise en compte dans le modèle, car elle détermine la détection des sujets de requête de faits et permet d'éviter le double comptage des données factuelles.

Le logiciel IBM Cognos utilise les règles de base suivantes pour appliquer la cardinalité lors de la génération des requêtes :

- La cardinalité est appliquée dans le contexte d'une requête.
- La cardinalité 1 à n implique des données de fait du côté n et des données de dimension du côté 1.
- Un sujet de requête peut se comporter en tant que sujet de requête de fait ou en tant que sujet de requête dimensionnel, selon les relations requises pour répondre à une requête particulière.

### Données fragmentées

Lors de la modélisation à des fins d'analyse ou de génération de rapports, il est important de prendre en compte la nature des questions métier par rapport à la nature de la source de données.

Dans un scénario courant, une relation entre une dimension et une table de faits dans un schéma en étoile est facultative. Cela signifie que tous les membres dimensionnels ne sont pas obligatoires dans la table des faits. Ceci est compensé par les moteurs OLAP en insérant une valeur appropriée lors de la création de la structure OLAP pour les points d'intersection dimensionnelle qui n'ont pas de données.

Par exemple, un utilisateur Analysis Studio souhaite créer ce rapport :

Région	2005	2006
Canada		1 000 000
Mexique	500 000	750 000
États-Unis	1000 000	1250 000

Lors de la modélisation, les relations facultatives entre les dimensions et les faits sont souvent remplacées pour améliorer les performances. Toutefois, lorsque vous effectuez une analyse ou une génération de rapports sur des données fragmentées dans lesquelles vous avez besoin d'informations sur les membres dimensionnels qui ne comportent pas de faits, les jointures externes doit être activées pour que des données soient renvoyées pour les points d'intersection dimensionnelle valides.

Pour activer les jointures externes, procédez comme suit :

- Vérifiez auprès de votre administrateur de bases de données que la source de données peut prendre en charge les jointures externes complètes.
- Importez les métadonnées avec les jointures externes activées.

## Modification d'une relation

Après avoir importé des données ou créé une relation dans IBM Cognos Framework Manager, vous pouvez renommer la relation et redéfinir la cardinalité.

Vous pouvez créer des expressions de relation personnalisée en sélectionnant un opérateur dans la liste ou en modifiant manuellement les expressions dans l'éditeur d'expression.

Vous pouvez également créer une expression complexe pour la relation.

Vous pouvez afficher les relations qui existent déjà pour un objet en sélectionnant ce dernier et en cliquant sur **Lancer Context Explorer** dans le menu **Outils**.


### Procédure

1. Cliquez sur une relation, puis sur **Actions, Editer la définition**.
2. Pour modifier des éléments existants, dans l'onglet **Expression de relation**, sélectionnez les éléments de requête, les cardinalités, et l'opérateur voulus.  
Les éléments de requête doivent avoir le même type de données.
3. Pour créer une jointure supplémentaire, dans l'onglet **Expression de relation**, cliquez sur **Nouveau lien** et définissez la nouvelle relation.
4. Pour tester la relation, dans l'onglet **Instruction SQL de relation**, indiquez le nombre de lignes à renvoyer, puis cliquez sur **Test**.
5. Cliquez sur **OK**.  
Si vos métadonnées proviennent d'une source de données OLAP, cliquez sur **Fermer**.

## Création d'expressions complexes pour une relation

Vous pouvez créer des expressions complexes pour des relations en utilisant des fonctions, des paramètres et des objets du modèle.

### Procédure

1. Cliquez sur une relation, puis sur **Actions, Editer la définition**.
2. Dans l'onglet **Expression de relation**, cliquez sur les points de suspension (...) en regard de la zone **Expression**.
3. Définissez l'expression.  
Si vous insérez des paramètres de session ou des invites et que vous souhaitez indiquer les valeurs qu'ils représentent lorsque vous testez l'expression, cliquez sur le bouton d'options .
4. Cliquez sur **OK**.

## Création de relations

Vous pouvez créer une relation pour joindre logiquement des objets associés que les utilisateurs souhaitent combiner dans un même rapport. Cela s'avère utile pour des relations entre des objets qui n'ont pas été sélectionnés lors de l'importation de métadonnées, qui n'ont pas été joints dans la source de données ou qui proviennent de plusieurs sources.

Vous pouvez créer directement une relation entre des éléments de requête.

Vous pouvez également créer une expression complexe pour la relation.

Vous pouvez aussi utiliser IBM Cognos Framework Manager pour générer automatiquement des relations entre des objets en fonction de critères sélectionnés.

Vous pouvez afficher les relations qui existent déjà pour un objet en sélectionnant ce dernier et en cliquant sur **Lancer Context Explorer** dans le menu **Outils**.

### Procédure

1. Appuyez sur Ctrl et cliquez sur une ou deux dimensions, ou sur un ou deux élément de requêtes.
2. Cliquez sur **Actions, Créer, Relations**.  
Si cette relation est une cible valide pour un raccourci de relation, Framework Manager vous demande si vous souhaitez créer un raccourci vers cette relation. Pour plus d'informations, voir «Création d'un raccourci de relation».
3. Cliquez sur **OK**.  
La boîte de dialogue **Définition de relation** s'affiche. Elle permet de modifier la relation.

## Création d'un raccourci de relation

Un raccourci de relation est un pointeur vers une relation existante. Vous pouvez utiliser des raccourcis de relation pour réutiliser la définition d'une relation existante. Les modifications apportées à la relation source sont automatiquement répercutées dans le raccourci. Vous pouvez également utiliser des raccourcis de relation pour résoudre des relations ambiguës entre des sujets de requête.

IBM Cognos Framework Manager vous demande si vous souhaitez créer un raccourci de relation lorsque vous créez une relation et que les deux conditions suivantes s'appliquent :

- Au moins une extrémité de la nouvelle relation est un raccourci.
- Une relation existe entre les objets d'origine.

### Procédure

1. Appuyez sur Ctrl et cliquez sur les objets qui doivent participer au raccourci de relation.
2. Cliquez sur **Actions, Créer, Relations**.  
Framework Manager vous demande si vous souhaitez créer un raccourci vers cette relation.
3. Cliquez sur **Oui**.  
Une liste affiche de toutes les relations dans lesquelles une extrémité est un objet de modèle, et l'autre extrémité est un autre objet de modèle ou un raccourci vers un autre objet de modèle.
4. Pour extraire toutes les relations dans lesquelles les deux extrémités peuvent être un objet de modèle ou un raccourci vers un objet de modèle, cliquez sur **Rechercher tout**.
5. Cliquez sur la relation devant être la cible du raccourci de relation.
6. Cliquez sur **OK**.

## Détection et génération de relations

Vous pouvez utiliser IBM Cognos Framework Manager pour détecter et générer des relations entre plusieurs objets existants de votre modèle. Cela s'avère utile

lorsque vous importez des métadonnées par étapes, ou lorsque vous souhaitez modifier les critères qui s'appliquent à des relations existantes, par exemple, l'inclusion ou non de jointures externes.

Lors de l'importation de métadonnées de schéma en étoile, évitez de générer des relations basées sur des noms de colonne ou d'élément de requête correspondants, sauf si des conventions de dénomination sont en vigueur. Les entrepôts de données appliquent souvent des normes de dénomination aux colonnes, par exemple, `surr_key` comme nom de colonne par défaut pour les clés secondaires dans des dimensions. Dans ce cas, la génération de relations basées sur des noms de colonne correspondants génère des relations incorrectes entre toutes les tables de dimension.

## Procédure

1. Appuyez sur **Ctrl** et cliquez sur plusieurs objets.
2. Cliquez sur **Outils, Détecter les relations**.
3. Sélectionnez les règles à appliquer à chaque paire de tables.

Règle	Résultat
Utiliser les clés primaires et externes	Crée des jointures basées sur des relations de clé primaire et externe. Les noms d'élément de requête n'ont pas de correspondance.
Utiliser les noms d'élément de requête correspondants qui représentent des colonnes indexées de manière unique	Crée des jointures entre des éléments de requête dont les noms et les types de données correspondent si l'une des colonnes sous-jacentes ou les deux sont indexées de manière unique.
Utiliser les noms des éléments de requête correspondants	Crée des jointures entre des éléments de requête dont les noms et les types de données correspondent. Cela génère autant de relations que possible.

4. Indiquez si vous souhaitez que Framework Manager détecte et génère des relations entre
  - Les objets sélectionnés
  - Chaque objet sélectionné et chaque objet du projet qui n'est pas sélectionné
  - Les objets sélectionnés et tout autre objet du projet
5. Déterminez si vous souhaitez que Framework Manager crée des jointures externes ou des jointures internes basées sur des jointures externes existant dans la source de données.
6. Si vous souhaitez désactiver la prévention automatique du double comptage, convertissez toutes les cardinalités `n en 1` en désélectionnant la case **Détection des faits activée**.  
Pour plus d'informations, voir Chapitre 9, «Guide de modélisation des métadonnées», à la page 327.
7. Cliquez sur **OK**.

---

## Sujets de requête

Un sujet de requête est un ensemble d'éléments de requête liés par une relation intrinsèque.

Vous pouvez utiliser IBM Cognos Framework Manager pour modifier les sujets de requête afin d'optimiser et de personnaliser les données qu'ils extraient. Par exemple, vous pouvez ajouter des filtres ou des calculs. Lorsque vous modifiez la définition d'un sujet de requête, Framework Manager régénère les éléments de requête associés, pour faire en sorte que toutes les modifications apportées aux propriétés du sujet de requête soient répercutées dans tous les éléments de requête pour ce sujet de requête.

Il existe différents types de sujet de requête dans Framework Manager :

- Sujets de requête de source de données
- Sujets de requête de modèle
- Sujets de requête de procédure stockée

Vous pouvez être amené à consulter «Sujets de requête et dimensions», à la page 341.

### Sujets de requête de source de données

Les sujets de requêtes qui sont basés sur des métadonnées relationnelles sont définies par des instructions SQL qui décrivent comment extraire les données de la source de données.

Les sujets de requête de source de données font directement référence à des données d'une source de données unique. IBM Cognos Framework Manager crée automatiquement un sujet de requête de source de données pour chaque table et vue que vous importez dans votre modèle.

Par exemple, vous importez la table de faits détaillés Employé à partir de l'exemple de base de données Entrepôt de données de Vacances et Aventure. Framework Manager crée alors un sujet de requête à l'aide de l'instruction SQL suivante :

```
Select * from [go_data_warehouse].EMPLOYEE_DETAIL_FACT
```

Framework Manager génère des sujets de requête qui représentent des données tabulaires issues de la source de données. Par exemple, un sujet de requête qui fait référence à la totalité d'une table contient des éléments de requête qui représentent chaque colonne de la table. Si l'instruction SQL sélectionne uniquement des colonnes spécifiques, seules ces colonnes sont représentées en tant qu'éléments de requête.

Chaque sujet de requête de source de données peut faire référence à des données d'une seule source de données à la fois. Toutefois, les sujets de requête de source de données ont pour avantage de vous permettre d'éditer directement le code SQL qui définit les données à extraire. En d'autres termes, vous pouvez insérer des paramètres permettant de contrôler les données que la requête extrait et créer des sujets de requête basés sur du code SQL arbitraire.

Vous pouvez avoir créé des instructions SQL pour des sujets de requête sans inclure de références à des colonnes et des tables dans des guillemets. Cela fonctionne dans les environnements de source de données relationnelles qui

n'utilisent pas une recherche sensible à la casse sur les tables de métadonnées analysées. L'instruction SQL pour le sujet de requête peut être du code SQL Cognos, SQL natif ou SQL direct. Les instructions SQL natives et directes doivent être totalement autonomes et ne doivent pas faire référence à des éléments externes à ce code SQL, par exemple, à des invites de base de données, des variables ou un formatage natif qui sont normalement fournis par l'application appelante. Les instructions SQL Cognos sont toutefois analysées à l'aide de métadonnées issues du modèle ou la source de données relationnelles. Par défaut, SQL Cognos est sensible à la casse. La recherche des métadonnées est donc effectuée en utilisant les noms tels qu'ils s'affichent dans l'instruction SQL.

Pour utiliser plusieurs sources de données pour un sujet de requête, faites appel à un sujet de requête de modèle qui accède aux sujets de requête de source de données ou à d'autres sujets de requête de modèle.

### **Création de sujets de requête de source de données**

Les sujets de requête de source de données font directement référence à des données d'une source de données unique. IBM Cognos Framework Manager crée automatiquement un sujet de requête de source de données pour chaque table et vue que vous importez dans votre modèle. Vous pouvez créer des sujets de requête de source de données supplémentaires.

Vous pouvez ajouter toute combinaison d'objets dans un sujet de requête, par exemple, des éléments de requête, d'autres sujets de requête ou des dimensions. Vous pouvez ajouter des calculs et des filtres autonomes et aussi incorporer des calculs et des filtres dans le sujet de requête.

Vous pouvez également créer des sujets de requête de modèle qui sont basés sur des métadonnées qui existent dans votre modèle, ainsi que des sujets de requête de procédure stockée qui sont générés à partir des procédures stockées d'une source de données relationnelles.

### **Procédure**

1. Sélectionnez le dossier des espaces-noms et cliquez sur **Actions, Créer, Sujet de requête**.
2. Dans la zone **Nom**, entrez un nom pour le nouveau sujet de requête.
3. Cliquez sur **Source de données, OK**.
4. Suivez toutes les étapes de l'assistant **Nouveau sujet de requête**.  
Pour vous assurer que la source de données est identifiée de façon unique pour un sujet de requête de source de données, ne quittez pas l'assistant avant que le bouton **Terminer** s'affiche.
5. Cliquez sur **Terminer**.
6. Cliquez avec le bouton droit de la souris sur le sujet de requête que vous avez créé et cliquez sur **Editer la définition**.
7. Cliquez sur l'onglet **SQL** et faites glisser des objets depuis la zone **Objets de base de données disponibles** vers la zone **SQL**.

Si votre projet contient plusieurs sources de données et que vous souhaitez ajouter au sujet de requête des objets provenant de différentes sources de données, cliquez sur **Insérer une source de données**, sélectionnez la source de données requise, puis cliquez sur **OK**.

Vous pouvez également insérer une macro et incorporer un calcul ou un filtre. Pour plus d'informations, voir «Création de calculs», à la page 152 ou «Création de filtres», à la page 155.

8. Choisissez une opération :

Objectif	Action
Fournir un contrôle sur la granularité	Cliquez sur l'onglet <b>Déterminants</b> . Voir «Déterminants», à la page 84.
Tester le sujet de requête	Cliquez sur l'onglet <b>Test</b> . Voir «Test de sujets de requête ou d'ensembles de requêtes», à la page 93.
Afficher le code SQL	Cliquez sur l'onglet <b>Informations sur la requête</b> . Voir «Edition du code SQL», à la page 99.
Afficher les tables système issues de la source de données	Cochez la case <b>Afficher les objets système</b> .

9. Cliquez sur **OK**.

Un avertissement s'affiche si des modifications ont invalidé des relations, d'autres sujets de requête, des calculs ou des filtres.

10. Vérifiez que les propriétés **Utilisation** et **Agrégat ordinaire** sont correctement définies. Voir «Modification du mode d'agrégation des éléments de requête», à la page 136.

## Sujets de requête de modèle

Les sujets de requête de modèle ne sont pas générés directement à partir d'une source de données, mais ils sont basés sur des éléments de requête d'autres sujets de requête ou dimensions, y compris des sujets de requête d'autres modèles. En utilisant des sujets de requête de modèle, vous pouvez créer une vue orientée métier plus abstraite d'une source de données.

En règle générale, les sujets de requête de modèle sont créés dans la vue métier, et non dans la vue d'importation. Pour plus d'informations sur la vue métier, voir «Organisation des modèles», à la page 176.

Les sujets de requête de modèle étant basés sur les métadonnées contenues dans votre modèle, ils vous permettent de :

- réutiliser des instructions SQL complexes contenues dans le modèle ;
- référencer des objets à partir de différentes sources de données dans le même sujet de requête.

Si vous importez un sujet de requête de modèle depuis un autre modèle, le sujet de requête de modèle ne fonctionnera pas sauf si vous importez également les sujets de requête de source de données référencés par le sujet de requête de modèle.

Si vous souhaitez modifier le code SQL, vous devez convertir le sujet de requête de modèle en un sujet de requête de source de données.



## Création de sujets de requête de modèle pour des métadonnées relationnelles

Les sujets de requête de modèle sont basés sur des éléments de requête contenus dans d'autres sujets de requête ou dimensions, y compris d'autres sujets de requête de modèle. En utilisant des sujets de requête de modèle, vous pouvez créer une vue orientée métier plus abstraite d'une source de données.

Vous pouvez ajouter toute combinaison d'objets dans un sujet de requête, par exemple, des éléments de requête, d'autres sujets de requête ou des dimensions. Vous pouvez ajouter des calculs et des filtres autonomes et aussi incorporer des calculs et des filtres dans le sujet de requête.

Vous pouvez créer un sujet de requête de modèle en fusionnant des sujets de requête et des éléments de requête existants («Création de sujets de requête de modèle basés sur des objets existants», à la page 88). Vous pouvez également créer des sujets de requête de source de données qui font directement référence à des données d'une source de données unique («Sujets de requête de source de données», à la page 76), ainsi que des sujets de requête de procédure stockée qui sont générés à partir des procédures stockées d'une source de données relationnelles («Sujets de requête de procédure stockée», à la page 80).

Lorsque vous utilisez un sujet de requête de modèle dans les studios IBM Cognos, IBM Cognos BI fait appel aux relations que vous avez définies pour le sujet de requête de modèle et non celles définies pour les sujets de requête de source de données sous-jacents. Si vous souhaitez que les relations définies pour les sujets de requête de source de données sous-jacents soient utilisées, ne définissez pas de relations ou de déterminants pour le sujet de requête de modèle.

### Procédure

1. Sélectionnez le dossier des espaces-noms et cliquez sur **Actions, Créer, Sujet de requête**.
2. Dans la zone **Nom**, entrez un nom pour le nouveau sujet de requête.
3. Cliquez sur **Modèle, OK**.
4. Cliquez sur l'onglet **Définition de sujet de requête**.
5. Pour ajouter des éléments à un sujet de requête de modèle, faites glisser les éléments depuis la zone **Objets de modèle disponibles** vers la zone **Éléments de requête et calculs**.

Vous pouvez changer l'ordre des éléments et des calculs. Toutefois, si le sujet de requête contient un dossier d'élément de requête, vous pouvez changer l'ordre des éléments uniquement dans l'**afficheur de projet**.

6. Vous pouvez également incorporer un filtre.  
Si vous ajoutez des calculs ou des filtres à un sujet de requête de modèle, Framework Manager doit accéder à la source de données au lieu d'accéder simplement au modèle.
7. Si vous souhaitez tester le sujet de requête, cliquez sur l'onglet **Test**.
8. Cliquez sur **OK**.

Un avertissement s'affiche si des modifications ont invalidé des relations, d'autres sujets de requête, des calculs ou des filtres.

9. Vérifiez que les propriétés **Utilisation** et **Agrégat ordinaire** sont correctement définies.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Calculs intégré : «Création d'un calcul (SAP BW)», à la page 240
- Filtres intégrés : «Création d'un filtre (SAP BW)», à la page 242
- Test et définition des options de test : «Test d'une dimension ou d'un autre objet (SAP BW)», à la page 215
- Modification des propriétés pour plusieurs sujets de requête en même temps : «Remplacement de plusieurs valeurs de propriété», à la page 17

## Sujets de requête de procédure stockée

Des sujets de requête de procédure stockée sont générés lorsque vous importez une procédure à partir d'une source de données relationnelles. IBM Cognos Framework Manager ne prend en charge que des procédures stockées définies par l'utilisateur. Les procédures stockées système ne sont pas prises en charge.

La procédure doit être exécutée dans Framework Manager pour obtenir une description de l'ensemble de résultats que la procédure peut renvoyer.

La procédure stockée doit renvoyer un ensemble de résultats uniforme unique. IBM Cognos BI prend en charge uniquement le premier ensemble de résultats renvoyé. Si la procédure peut éventuellement renvoyer un autre ensemble de résultats, son format doit être cohérent avec celui utilisé pour définir les métadonnées dans Framework Manager.

Chaque ensemble de résultats doit renvoyer le même formulaire, avec le même nombre, les mêmes types et les mêmes noms de colonnes par exemple. Les signatures surchargées sont prises en charge par IBM Cognos BI, mais chaque procédure doit être définie avec un nom unique et un sujet de requête distinct pour chaque ensemble de résultats.

Les paramètres de sortie ne sont pas pris en charge.

Une fois que vous avez importé ou créé un sujet de requête de procédure stockée, il s'affiche sous la forme d'un objet. Vous devez l'exécuter pour valider la procédure stockée sous-jacente et spécifier la liste de projection. Souvent, des métadonnées statiques n'existent pas pour la procédure stockée dans la source de données relationnelles qui décrit la présentation admise des ensembles de résultats. L'ensemble de résultats peut n'être connu que lors de l'exécution. Lorsqu'une procédure stockée est mise à jour dans la source de données, l'exécution de la procédure stockée dans Framework Manager met à jour le sujet de requête à l'aide des éléments de requête nouvellement générés.

Parfois, des fonctions sont importées en tant que sujets de requête de procédure stockée. Vérifiez la définition de procédure stockée afin de déterminer les valeurs transmises attendues par celle-ci et ce qu'elle tente de renvoyer. Editez et testez chaque sujet de requête de procédure stockée dont vous pensez qu'il peut s'agir d'une fonction. Si le test échoue, le sujet de requête est une fonction et doit être supprimé.

Pour plus d'informations, voir «Création ou modification de sujets de requête de procédure stockée», à la page 81.

## Procédures stockées issues de sources de données Informix

Si vous disposez de procédures stockées issues de sources de données Informix Dynamic ou Parallel Server, vous devez modifier les paramètres. Reportez-vous à la source d'origine des procédures stockées pour vous assurer qu'elles sont correctement mappées.

Informix 7.x et 8.x fournissent uniquement le nom de la procédure stockée à Framework Manager. Vous devez définir tous les paramètres, tels que le nom de paramètre, le type de données, le mode, la taille, la précision, l'échelle et une valeur pour qu'un ensemble de résultats puisse être obtenu. Informix 9.x fournit des métadonnées pour les procédures stockées et les fonctions définies par l'utilisateur avec des valeurs de paramètre par défaut. Vérifiez tous les paramètres avant de les utiliser, notamment l'attribut de mode.

Les fonctions Informix sont importées en tant que procédures stockées. Après les avoir importées, vous devez les changer en fonctions en cliquant sur le bouton **f(x)** dans la boîte de dialogue **Editer la définition**. Ce bouton n'est activé que pour ces fonctions. Sélectionnez ensuite l'argument qui représente les résultats ou utilisez les valeurs obtenues à partir des résultats de test.

## Procédures stockées issues de Composite Information Server

Si vous disposez de procédures stockées issues de Composite Information Server, reportez-vous à la source d'origine des procédures stockées pour vous assurer qu'elles sont correctement mappées.

Les fonctions Composite sont importées en tant que procédures stockées. Après les avoir importées, vous devez les changer en fonctions en cliquant sur le bouton **f(x)** dans la boîte de dialogue **Editer la définition**. Ce bouton n'est activé que pour ces fonctions. Sélectionnez ensuite l'argument qui représente les résultats ou utilisez les valeurs obtenues à partir des résultats de test.

## Création ou modification de sujets de requête de procédure stockée

Une fois que vous avez importé ou créé un sujet de requête de procédure stockée, vous pouvez le modifier. Pour éviter les incohérences, le sujet de requête modifié doit renvoyer la même structure d'ensemble de résultats que la procédure stockée d'origine.

IBM Cognos Framework Manager ne prend en charge que les procédures stockées définies par l'utilisateur. Les procédures stockées système ne sont pas prises en charge.

Il existe différents types de procédure stockée :

Type de procédure stockée	Description
Requête de données	<p>Emet une transaction en lecture seule</p> <p>Si vous disposez d'une procédure stockée dont le type est défini sur <b>Requête de données</b>, celle-ci génère une transaction en lecture seule. Lorsque vous exécutez la procédure stockée dans Event Studio, un message d'erreur indique que la procédure stockée veut mettre à jour la base de données. La cause de cette erreur est que la procédure stockée contient une transaction passive qui est prise en charge par la base de données sous-jacente. La solution consiste à cliquer sur <b>OK</b> pour que la procédure stockée mette à jour la base de données. Aucune autre action n'est requise.</p>
Modification de données	<p>Ecrit un enregistrement dans la source de données. Utilisez ce type si vous souhaitez employer la procédure stockée dans Event Studio.</p> <p>Si vous souhaitez que les utilisateurs Event Studio puissent sélectionner un paramètre dans une tâche, vous devez placer le paramètre entre guillemets.</p> <p><b>Avertissement :</b> Le test d'une procédure stockée de modification de données dans la boîte de dialogue <b>Editer la définition</b> se traduit par l'écriture des données dans la source de données. Vous ne pouvez pas annuler des transactions effectuées sur la source de données dans Framework Manager. Si des données indésirables sont écrites dans la source de données suite au test de la procédure stockée, une annulation peut être effectuée par l'administrateur de base de données si la source de données est configurée pour prendre en charge cette annulation. Pour tester la procédure stockée sans que des données soient écrites dans la source de données, cliquez sur <b>Test</b> dans le menu <b>Outils</b>.</p>

Vous pouvez également créer des sujets de requête de source de données qui font directement référence à des données d'une source de données unique («Sujets de requête de source de données», à la page 76), ainsi que des sujets de requête de modèle qui sont basés sur des métadonnées existant dans votre modèle («Sujets de requête de modèle», à la page 78).

## Procédure

1. Procédez comme suit :

Objectif	Action
Créer un sujet de requête de procédure stockée	Sélectionnez le dossier des espaces-noms et, dans le menu <b>Actions</b> , cliquez sur <b>Créer, Sujet de requête</b> .  Dans la zone <b>Nom</b> , entrez un nom pour le nouveau sujet de requête.  Cliquez sur <b>Procédure stockée</b> , puis sur <b>OK</b> .  Suivez toutes les étapes de l'assistant <b>Nouveau sujet de requête</b> .
Modifier un sujet de requête de procédure stockée	Sélectionnez le sujet de requête de procédure stockée à modifier.  Dans le menu <b>Actions</b> , cliquez sur <b>Editer la définition</b> .

2. Cliquez sur l'onglet **Définition** et sélectionnez l'action voulue.

Objectif	Action
Utiliser une procédure stockée différente	Dans la zone <b>Nom de la procédure stockée</b> , saisissez le nom de la procédure stockée.
Modifier le type de la procédure stockée	Dans la zone <b>Type</b> , sélectionnez <b>Requête de données</b> ou <b>Modification de données</b> .
Modifier la source de données dans laquelle se trouve la procédure stockée	Cliquez sur les points de suspension (...) en regard de la zone <b>Source de données</b> .  Lorsque vous importez une procédure stockée, une nouvelle source de données est créée. Vous pouvez pointer vers la source de données d'origine et supprimez la nouvelle source de données.
Editer un argument	Cliquez sur l'argument, puis sur les points de suspension (...).  La zone <b>Syntaxe</b> de la boîte de dialogue <b>Définition de sujet de requête</b> indique la syntaxe correcte à utiliser.
Générer l'élément de requête projeté	Cliquez sur l'onglet <b>Test</b> . Voir «Test de sujets de requête ou d'ensembles de requêtes», à la page 93.

3. Cliquez sur **OK**.

Framework Manager exécute la procédure stockée et valide le sujet de requête si ce dernier renvoie un ensemble de résultats.

Si la procédure stockée ne renvoie pas d'ensemble de résultats, le sujet de requête devient un sujet de requête non valide s'il est enregistré dans le modèle. Si le sujet de requête non valide est inclus dans le pack publié, il ne peut pas être utilisé dans un rapport.

4. Vérifiez que les propriétés **Utilisation** et **Agrégat ordinaire** sont correctement définies pour chaque élément de requête nouvellement créé.

Par exemple, un élément de requête peut être défini en tant que fait alors qu'il s'agit d'un identificateur.

## Résultats

Vous pouvez mettre à jour le sujet de requête de procédure stockée si la source de données est modifiée. Voir «Mise à jour de sujets de requête», à la page 97.

### Exemple - Utilisation d'invites avec une procédure stockée

Si vous définissez des invites pour des variables de procédure stockée, les utilisateurs peuvent définir les variables dans les rapports.

## Procédure

1. Créez un sujet de requête de procédure stockée qui utilise la procédure stockée `sp_FIND_ORDER_DATE`.

La boîte de dialogue **Définition de sujet de requête** s'affiche.

2. Dans l'onglet **Définition**, sélectionnez l'argument `@order_number`, puis cliquez sur les points de suspension (...).
3. Dans la zone **Valeur**, entrez la macro suivante, puis cliquez sur **OK**:

```
#prompt('Order Number','integer')#
```

**Remarque** : Framework Manager supprime tous les éléments figurant en dehors des signes dièse lors de l'exécution de la macro.

4. Si vous souhaitez tester l'invite pour la variable, procédez comme suit :

- Cliquez sur **Test, Tester l'exemple**.

La boîte de dialogue **Valeurs d'invites** s'affiche.

- Dans la colonne **Nom**, cliquez sur Numéro de commande.
- Dans la colonne **Valeur**, saisissez **1234** et cliquez sur **OK**.

Un enregistrement est renvoyé, indiquant la date pour le numéro de commande 1234.

Framework Manager utilise cette valeur pendant la durée de la session en cours ou jusqu'à ce que vous désélectionnez la valeur de l'invite.

5. Cliquez sur **OK**.

## Déterminants

Les déterminants reflètent la granularité en représentant des sous-ensembles ou des groupes de données dans un sujet de requête et ils permettent de garantir l'agrégation correcte de ces données répétées. Les déterminants sont étroitement liés au concept de clés et d'index de la source de données et sont importés en fonction des informations de clé et d'index uniques contenues dans cette source de données. Il est conseillé de toujours vérifier les déterminants importés et, si nécessaire, de modifier ces derniers ou d'en créer d'autres. Lorsque vous modifiez les déterminants, vous pouvez redéfinir les informations de clé et d'index contenues dans votre source de données, en les remplaçant par des informations qui correspondent mieux à vos besoins en matière d'analyse et de génération de

rapports. En ajoutant des déterminants, vous pouvez représenter des groupes de données répétées qui sont pertinentes pour votre application.

Voici un exemple de déterminant unique, Jour, dans l'exemple Temps ci-dessous. L'exemple de déterminant non unique est Mois ; la clé dans Mois est répétée selon le nombre de jours d'un mois donné. Lorsque vous définissez un déterminant non unique, vous devez spécifier **Grouper par**. Cela indique à IBM Cognos que lorsque les clés ou attributs associés à ce déterminant sont répétés dans les données, il convient d'appliquer des fonctions d'agrégation et un groupement afin d'éviter le double comptage. Il n'est pas conseillé de spécifier des déterminants en sélectionnant à la fois **Identifié de manière unique** et **Grouper par** ou aucune de ces deux valeurs.

Clé Année	Clé Mois	Nom du mois	Clé Jour	Nom du jour
2006	200601	6 janvier	20060101	Dimanche 1er janvier 2006
2006	200601	6 janvier	20060102	Lundi 2 janvier 2006

Vous pouvez définir trois déterminants pour ce fichier de la façon suivante : deux déterminants **Regrouper par** (Année et Mois) et un déterminant unique (Jour). Ce concept est similaire mais pas identique au concept de niveaux et de hiérarchies.

Nom du déterminant	Clé	Attributs	Identifié de manière unique	Grouper par
Année	Clé Année	Aucun	Non	Oui
Mois	Clé Mois	Nom du mois	Non	Oui
Jour	Clé Jour	Nom du jour Clé Mois Nom du mois Clé Année	Oui	Non

Dans ce cas, une seule clé est utilisée pour chaque déterminant car chaque clé contient suffisamment d'informations pour identifier un groupe parmi les données. Le sujet Mois est souvent difficile à gérer lorsque la clé ne contient pas assez d'informations pour indiquer clairement à quelle année le mois appartient. Dans ce cas, cependant, la clé Mois contient la clé Année qui à elle seule permet d'identifier les mois en tant que sous-groupe des années.

**Remarque :** Vous pouvez créer un déterminant qui regroupe les mois sans le contexte des années, mais ce choix est peu courant car le rapport correspondant contiendrait toutes les données du mois de février de toutes les années regroupées, au lieu des seules données de février 2006.

## Situations dans lesquelles utiliser des déterminants

Les déterminants peuvent servir à résoudre différents problèmes liés à la granularité des données, vous devez les utiliser systématiquement dans les principaux cas suivants :

- Un sujet de requête qui se comporte comme une dimension a plusieurs niveaux de granularité et sera joint à des données de faits sur différents ensembles de clés.

Par exemple, le sujet Temps a plusieurs niveaux et il est joint à Inventaire sur la clé Mois et à Ventes sur la clé Jour. Pour plus d'informations, voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

- Il est nécessaire d'effectuer un comptage ou d'autres fonctions d'agrégation sur une clé ou un attribut qui est répété.

Par exemple, Temps a une clé Mois et un attribut, Nombre de jours dans le mois, qui est répété pour chaque jour. Si vous voulez utiliser l'attribut Nombre de jours dans le mois dans un rapport, vous ne souhaitez pas obtenir la somme des jours du mois pour chaque jour du mois. En revanche, vous voulez obtenir la valeur unique de Nombre de jours dans le mois pour la clé Mois choisie. Dans SQL, cela donne XMIN(Days in the month for Month\_Key). Il existe également une clause Group by dans le code SQL Cognos.

Il existe quelques cas moins fréquents dans lesquels il convient d'utiliser des déterminants :

- Vous souhaitez identifier de façon unique la ligne de données lors de l'extraction de données texte BLOB de la source de données.

L'utilisation de requêtes sur des données BLOB requiert des informations de clé ou d'index supplémentaires. Si ces informations ne sont pas disponibles dans la source de données, vous pouvez les ajouter à l'aide de déterminants. Remplacez les déterminants importés depuis la source de données qui sont en conflit avec des relations établies pour la génération de rapports.

Il n'est pas possible d'utiliser des clés à segments multiples lorsque le sujet de requête accède aux données BLOB. Avec des requêtes récapitulatives, les données BLOB doivent être extraites séparément de la partie récapitulative de la requête. Pour cela, vous avez besoin d'une clé identifiant la ligne de façon unique et n'étant pas associée à plusieurs segments.

- Il existe une jointure définie qui utilise moins de clés qu'un déterminant unique défini pour un sujet de requête.

Si votre jointure est basée sur un sous-ensemble des colonnes qui sont référencées par les clés d'un déterminant unique du côté 0..1 ou 1..1 des relations, un conflit se produit. Pour résoudre ce conflit, modifiez la relation pour qu'elle soit parfaitement conforme au déterminant ou modifiez le déterminant de sorte qu'il prenne en charge la relation.

- Vous souhaitez remplacer les déterminants importés depuis la source de données qui sont en conflit avec des relations établies pour la génération de rapports.

Par exemple, il existe des déterminants pour deux sujets de requête correspondant à plusieurs colonnes, mais la relation entre ces sujets de requête n'utilise qu'un sous-ensemble de ces colonnes. Modifiez les informations des déterminants du sujet de requête si elles ne sont pas adaptées à l'utilisation de colonnes supplémentaires dans la relation.

## Spécification de déterminants

Les déterminants fournissent un contrôle sur la granularité pour les sujets de requête.

Si un sujet de requête comprend des déterminants, chaque élément de requête du sujet de requête doit être inclus dans l'un des déterminants.

Les déterminants sont traités dans l'ordre dans lequel ils sont indiqués dans le modèle. Vous pouvez modifier l'ordre des déterminants. Si un sujet de requête



contient plusieurs déterminant, le premier qui couvre tous les éléments requis est utilisé. Les déterminants sont évalués dans le contexte de chaque jointure requise, ainsi que dans le contexte des éléments requis.

Les sujets de requête de source de données sont importés avec des déterminants définis pour eux. Ces déterminants par défaut sont générés en fonction de clés et d'index de la source de données.

Des déterminants ne sont pas définis automatiquement pour les sujets de requête de modèle. Si des déterminants sont nécessaires, vous devez les définir manuellement.

Les sujets de requête de procédure stockée n'ont pas de déterminants.

Vous ne pouvez pas utiliser de déterminants avec du code SQL saisi par l'utilisateur, spécifié dans une requête définie dans Report Studio.

### Procédure

1. Cliquez sur le sujet de requête souhaité, puis cliquez sur **Actions, Editer la définition**.

2. Cliquez sur l'onglet **Déterminants**.

3. Cliquez sur **Ajouter** sous la zone **Déterminants**.

L'entrée **Nouveau déterminant** s'affiche dans la zone. Pour donner à cette entrée un nom pertinent, cliquez avec le bouton droit de la souris sur celle-ci, puis cliquez sur **Renommer**.

4. Pour définir une clé, cliquez avec le bouton droit sur un élément de requête dans la zone **Éléments disponibles**, puis cliquez sur **Ajouter en tant que clé**.

**Astuce** : Vous pouvez également faire glisser des éléments de requête vers la zone **Clé**.

5. Pour identifier les éléments de requête qui doivent être associés à ce déterminant, cliquez avec le bouton droit de la souris sur des éléments de requête dans la zone **Éléments disponibles** et cliquez sur **Ajouter en tant qu'attributs**.

**Astuce** : Vous pouvez également faire glisser des éléments de requête vers la zone **Attributs**.

Vous pouvez avoir un déterminant sans attributs définis pour celui-ci.

Framework Manager utilise ce type de déterminant pour indiquer les éléments de requête qui sont indexés.

6. Pour indiquer que le déterminant sélectionné doit être utilisé comme identificateur unique, cochez la case **Identification de façon unique**.

Effectuez cette opération uniquement si les données de cet élément sont uniques pour chaque ligne de la source de données sous-jacente.

Vous pouvez définir plusieurs déterminants uniques s'ils sont réellement uniques. Au moment de l'exécution de la requête, la relation utilisée détermine le déterminant unique à employer.

7. Cochez la case **Regroupement** pour indiquer que, lorsque les clés ou attributs associés à ce déterminant sont répétés dans les données, IBM Cognos BI doit appliquer des fonctions d'agrégat et un groupement afin d'éviter le double comptage.

8. Si vous souhaitez modifier l'ordre des déterminants, utilisez les flèches.

Les déterminants sont traités dans l'ordre dans lequel ils sont indiqués dans le modèle.

9. Cliquez sur OK.

## Résultats

Pour plus d'informations, voir «Déterminants», à la page 331 et Chapitre 10, «Code SQL généré par IBM Cognos Software», à la page 365.

### Effet des déterminants sur SQL

Il est essentiel de comprendre l'effet que produisent les déterminants sur le code SQL qui est généré. Les déterminants affectent le groupement et l'agrégation de données, y compris les autres sujets de requête qui ont des relations avec le sujet de requête et le sujet de requête lui-même.

Lisez par exemple les informations suivantes. Chaque sujet Ligne de produits contient de nombreuses occurrences de Type de produit. Chaque sujet Type de produit contient de nombreuses occurrences de Produit. Pour le sujet Produit, le sujet Clé de produit est une clé secondaire et le sujet Numéro de produit est une clé métier qui est utilisée comme un attribut. Les données jointes sur Clé de produit sont agrégées correctement lorsqu'elles sont signalées par Ligne de produits et/ou Type de produit.

Déterminant	Clé	Grouper par	Identifié de manière unique	Attributs
Ligne de produits	Cd ligne de produits	Oui		Ligne de produits
Type de produit	Cd type de produit	Oui		Type de produit
Produit	Clé de produit		Oui	Coût Marge Nom de produit Numéro de produit

## Création de sujets de requête de modèle basés sur des objets existants

Vous pouvez sélectionner des objets de modèle existants et les fusionner dans un nouveau sujet de requête de modèle. Vous pouvez ainsi réutiliser des métadonnées existantes pour créer rapidement des sujets de requête.

Des informations sur les sujets de requête de modèle SAP BW figurent dans une autre rubrique : «Création de sujets de requête de modèle basés sur des objets existants (SAP BW)», à la page 219.

### Pourquoi et quand exécuter cette tâche

Vous pouvez notamment fusionner les objets suivants :

- Sujets de requête de source de données relationnelles et leurs raccourcis
- Sujets de requête de modèle et leurs raccourcis
- Eléments de requête, filtres et calculs dans des sujets de requête de modèle et de source de données

- Relations et raccourcis de relation entre des sujets de requête de modèle et de source de données

Vous pouvez fusionner un nombre quelconque d'objets de même type dans une nouvelle requête en une seule opération. La fusion crée toujours un nouveau sujet de requête de modèle.

Le nouveau sujet de requête contient les filtres qui existaient dans le sujet de requête d'origine.

### Procédure

1. Appuyez sur Ctrl et cliquez sur les objets à fusionner dans un sujet de requête unique.
2. Cliquez sur **Actions, Fusionner dans le nouveau sujet de requête.**




## Affichage d'objets associés

Vous pouvez masquer un objet dans **Context Explorer**. Vous pouvez aussi modifier la présentation, ajuster tous les objets dans **Context Explorer**, effectuer un zoom avant ou arrière, imprimer, afficher un aperçu des diagrammes avant impression et modifier la mise en page.

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour explorer les dimensions.

### Procédure

1. Sélectionnez un ou plusieurs objets à explorer.
2. Dans le menu **Outils**, cliquez sur **Lancer Context Explorer**.
3. Pour afficher les objets connectés, cliquez sur un ou plusieurs objets, puis sur le bouton approprié.

Objectif	Bouton
Afficher les objets qui sont associés à l'objet sélectionné.	
Afficher les références immédiates pour les objets.	
Afficher toutes les références pour les objets.	

4. Si vous souhaitez afficher des détails sur un objet, par exemple, ses relations et ses éléments de requête, cliquez avec le bouton droit de la souris sur l'objet, cliquez sur **Parcourir le diagramme**, **Paramètres de diagramme**, puis sélectionnez les détails voulus.

## Création d'ensembles de requêtes

Tous les types de données ne sont pas pris en charge. En règle générale, les ensembles ne sont pas autorisés sur les types de données BFILE, BLOB, CLOB, LONG et VARRAY ou sur les colonnes de table imbriquées.

Un sujet de requête peut être défini à l'aide des opérations set d'union, d'intersection ou d'exception. Vous définissez un ensemble de requêtes pour fusionner, comparer ou associer des données similaires à partir de différentes sources de données. Les ensembles de requêtes sont utiles lors de la modélisation de données à partir de systèmes disparates.

Un certain nombre de raisons justifient la création d'un ensemble de requêtes. Un ensemble de requêtes peut s'avérer nécessaire pour créer une dimension conforme entre des sources de données disparates. Ou, vous pouvez être amené à comparer le contenu de deux requêtes pour déterminer si elles contiennent les mêmes données ; il s'agit d'une pratique courante dans les environnements de test. Ou, vous pouvez être amené à comparer des requêtes qui renvoient des valeurs nulles. Ou, vous souhaitez gérer une relation de type fait-fait qui est vraiment une relation de type 1-1. (S'il ne s'agit pas véritablement d'une relation de type 1-1, créez une le, créez une requête à granularité multiple «Requêtes à granularité multiple et à faits multiples», à la page 335.)

Un ensemble de requêtes peut contenir deux sujets de requête seulement. Vous pouvez créer un ensemble de requêtes qui fusionne deux autres ensembles de requêtes. Un ensemble de requêtes peut contenir :

- Toutes les lignes de deux sujets de requête (opération d'union).  
Par exemple, votre société a récemment fait l'acquisition d'une autre société et vous avez besoin de la liste complète de tous les clients.
- Seules les lignes partagées par les sujets de requête (opération d'intersection).  
Par exemple, vous souhaitez retrouver les membres du personnel qui sont également des responsables.
- Seules les lignes figurant dans le premier sujet de requête et non dans le second sujet de requête dans l'ensemble de requêtes (opération d'exception).  
Par exemple, vous souhaitez mettre en évidence les différences relatives à l'endroit où vos produits ont été vendus cette année par rapport à il y a dix ans.

Les noms des éléments de la liste de projection prennent par défaut les noms des éléments affectés au premier sujet de requête dans l'opération d'ensemble.

Les relations entre les deux sujets de requête de l'ensemble de requêtes et d'autres sujets de requête ne sont pas incluses dans l'ensemble de requêtes.

Différents résultats sont affichés dans les rapports selon l'opérateur utilisé. Par exemple, vos deux sujets de requête comportent les noms de divers employés.

Le premier sujet de requête contient les lignes suivantes :

Ligne	Valeur
1	Jane
2	John
3	John
4	Michael
5	Michael

Le second sujet de requête contient les lignes suivantes :

Ligne	Valeur
1	Jane
2	John
3	John
4	Patrick

Vous créez un ensemble de requêtes. Des résultats différents seront affichés en fonction de l'opérateur que vous utilisez.

Opérateur	Résultat	Remarques
Union	Jane, John, Michael, Patrick	Tous les éléments sont affichés. Les valeurs ne sont pas dupliquées.
Intersection	Jane, John	Les éléments communs sont affichés. Les valeurs ne sont pas dupliquées.
Exception	Michael	Les éléments qui ne sont pas communs sont affichés. Les valeurs ne sont pas dupliquées.  Si le second sujet de requête était défini en premier dans l'ensemble de requêtes, le résultat obtenu serait Patrick.
Union All	Jane, Jane, John, John, John, John, Michael, Michael, Patrick	Tous les éléments sont affichés. Les valeurs sont dupliquées.
Intersect All	Jane, John, John	Les éléments communs sont affichés. Les valeurs sont dupliquées.
Except All	Michael, Michael	Les éléments qui ne sont pas communs sont affichés. Les valeurs sont dupliquées.  Si le second sujet de requête était défini en premier dans l'ensemble de requêtes, le résultat obtenu serait Patrick.

## Procédure de création d'un ensemble de requêtes

### Procédure

1. Sélectionnez deux sujets de requête répondant aux exigences suivantes :
  - Chaque sujet de requête doit comporter le même nombre de colonnes.
  - Les colonnes doivent être dans le même ordre.
  - Les colonnes doivent avoir des types de données identiques ou similaires.

Les types de données n'ont pas besoin d'être exactement les mêmes si ceux du deuxième ensemble de résultats peuvent être automatiquement convertis par la source de données en types de données compatibles avec ceux du premier ensemble de résultats.

Par exemple, un sujet de requête contient des données de pays et utilise int comme type de données. Un autre sujet de requête contient des données pays et utilise small int comme type de données. Framework Manager importe ces sujets de requête en tant que int16 et int32 et effectue une opération set.
2. Cliquez sur **Actions, Définir l'ensemble de requêtes**.
3. Cliquez sur l'onglet **Définition**.
4. Dans la zone **Nom**, saisissez un nom pour l'ensemble de requêtes.
5. Vérifiez les zones **Sujet de requête** pour vérifier que l'ordre dans lequel les sujets de requête s'afficheront dans la clause Select est correct.

L'ordre peut être important si vous souhaitez qu'un ensemble spécifique de noms de colonne (alias) s'affiche dans un seul des sujets de requête. Si l'ordre est incorrect, annulez cet ensemble de requêtes et recommencez.

Pour les opérations union et intersect, l'ordre des sujets de requête n'a pas d'importance. Vous pouvez modifier l'ordre et obtenir le même résultat. Pour l'opération except, l'ordre des sujets de requête joue un rôle important.
6. Utilisez la zone **Opérateur** pour définir la manière dont les lignes des sujets de requête sont combinées.

Option	Description
Union	Extrait toutes les lignes uniques pour les deux ensembles. Les valeurs en double sont supprimées.
Intersect	Extrait toutes les lignes qui sont communes aux deux sujets de requête.
Except	Extrait les lignes qui existent dans le premier sujet de requête, mais pas dans le deuxième.

7. Pour créer une opération Union All, Intersect All ou Except All, désélectionnez la case **Retirer les lignes en double**.
8. Sélectionnez l'action voulue.

Objectif	Action
Travailler avec des calculs intégrés dans des sujets de requête	Cliquez sur l'onglet <b>Calculs</b> .  Vous pouvez ajouter ou éditer des calculs et changer leur ordre.
Travailler avec des filtres intégrés dans des sujets de requête	Cliquez sur l'onglet <b>Filtres</b> .  Vous pouvez ajouter ou éditer des filtres, et modifier leur ordre et leur utilisation.

Objectif	Action
Tester le sujet de requête	Cliquez sur l'onglet <b>Test</b> .

9. Cliquez sur **OK**.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Calculs intégrés : «Création de calculs», à la page 152
- Filtres intégrés : «Création de filtres», à la page 155
- Déterminants : «Déterminants», à la page 84
- Test de l'ensemble de requêtes et modification des paramètres de test : «Test de sujets de requête ou d'ensembles de requêtes»

## Test de sujets de requête ou d'ensembles de requêtes

### Test d'objets

Vous pouvez voir les résultats renvoyés par un objet en le testant. Vous pouvez tester un objet lorsque vous le créez ou ultérieurement. Les objets que vous pouvez tester sont les dimensions, les sujets de requête, les ensembles de requêtes, les hiérarchies, les niveaux, les calculs, et les éléments de requête.

Vous pouvez visualiser les données qui seront affichées dans un rapport spécifique avant de publier un pack en sélectionnant et testant les objets qui apparaîtront dans le rapport. Il est ainsi plus facile de déboguer un modèle et de vérifier que celui-ci répond aux exigences des rapports, car vous n'avez pas besoin de créer et de publier d'abord des packs.

Lorsque vous testez un objet, IBM Cognos Framework Manager renvoie des exemples de données. Aucun formatage n'est appliqué aux exemples de données. Si vous devez tester un formatage, vous devez publier le pack et afficher les objets dans les studios IBM Cognos.

Vous pouvez voir différents résultats en fonction de ce que vous testez. Par exemple, si vous utilisez l'éditeur d'expression pour tester un calcul qui est intégré dans un sujet de requête, Framework Manager teste uniquement l'expression, pas l'élément. Le paramètre d'agrégation pour l'élément de requête n'est donc pas appliqué au test. Le test de la totalité du sujet de requête, qui inclut le calcul, donne un résultat différent car le paramètre d'agrégation est appliqué. Par exemple, si le paramètre d'agrégation a pour valeur Récapituler, vous verrez moins de lignes dans le test.

Lorsque vous testez une dimension de mesure, SQL utilise des agrégats, pas les mesures.

Si vous testez un segment enfant d'un modèle segmenté, une erreur peut s'afficher si un objet que vous testez fait référence à un objet d'un autre segment enfant et que l'objet référencé n'est pas disponible pour le projet en cours. Vérifiez que le modèle parent contient tous les objets et que ce message d'erreur ne s'affiche pas lorsque vous testez le modèle parent.

Des paramètres de restriction peuvent affecter les résultats de test. Pour plus d'informations, voir «Restrictions», à la page 309.

Vous pouvez modifier les paramètres de test existants pour personnaliser les résultats affichés par le test. Par exemple, en plus d'autres paramètres, vous pouvez contrôler le nombre de lignes renvoyées.

## Etapes lors de la création ou de la modification de l'objet

### Procédure

1. Sélectionnez l'objet à tester.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.  
La zone **Résultats du test** est initialement vide tant que vous n'avez pas exécuté la requête.  
Tous les ensembles de résultats qui contiennent des objets BLOB (Binary Large Object) sont affichés en tant que [blob].
3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Si vous souhaitez ajouter un nombre de lignes, cliquez sur **Nombre total de lignes**.
5. Si vous souhaitez appliquer la propriété **Agrégat ordinaire** de l'élément de requête ou la propriété **Agrégat des règles** d'une mesure de semi-additive qui est référencée dans l'expression, cochez la case **Total automatique**.  
Si vous désactivez cette case, une ligne est renvoyée pour chaque ligne de l'ensemble de résultats de la requête.
6. Si vous souhaitez obtenir plus d'informations sur les résultats de requête, cliquez sur l'onglet **Informations sur la requête**.
7. Cliquez sur **OK**.

### Procédure de visualisation des données qui seront affichées dans un rapport spécifique

#### Procédure

1. Sélectionnez les objets qui s'afficheront dans le rapport.
2. Cliquez sur **Outils, Test**.
3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Pour afficher des détails sur tout problème trouvé, cliquez sur l'onglet **Informations sur la requête**.  
Si vous ne voyez pas les résultats de la requête dans la fenêtre de test, il se peut que les données de votre source de données dépassent la valeur de l'une des restrictions. La requête s'arrête à la limite spécifiée, mais la fenêtre des résultats de test ne contient aucune donnée. **Astuce** : Définissez chaque restriction sur zéro.

### Modification des paramètres de test

Vous pouvez personnaliser les tests en modifiant les paramètres de test.

#### Procédure

1. Sélectionnez l'objet de votre choix.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.
3. Cliquez sur **Options, Tester les paramètres**.
4. Sélectionnez les options de votre choix.



Objectif	Action	Persistance
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes.</p> <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>	<p>Ce paramètre s'applique à toutes les dimensions, tous les sujets de requête et tous les ensembles de requêtes du modèle.</p> <p>Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.</p>
Indiquer le niveau de détail	Faites glisser le curseur de <b>Niveau de détail affiché dans les informations de résultat</b> pour représenter la quantité d'informations dont vous avez besoin.	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.
Remplacer temporairement des paramètres de session	<p>Dans la zone <b>Paramètres de session</b>, cliquez sur <b>Définir</b>.</p> <p>La boîte de dialogue <b>Paramètres de session</b> s'affiche.</p>	Ces valeurs de remplacement ne sont pas enregistrées avec le modèle. Ce paramètre est disponible uniquement pour la session en cours.
Appliquer les filtres de mode de conception appropriés	<p>Cochez la case <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b>.</p> <p>Ce paramètre s'applique à tous les filtres pertinents dont l'utilisation est définie sur le mode de conception dans une autre dimension, un autre sujet de requête ou un autre ensemble de requêtes.</p>	Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.
Appliquer un filtre de sécurité	Dans la zone <b>Filtres de sécurité</b> , cliquez sur <b>Editer</b> .	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.
Modifier les valeurs d'invites	<p>Dans la zone <b>Valeurs d'invite en cours</b>, cliquez sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>	<p>Les valeurs d'invite ne sont pas enregistrées avec le modèle.</p> <p>Ce paramètre est disponible uniquement pour la session en cours.</p>

5. Cliquez sur le bouton **OK** à deux reprises.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Définition de restrictions
- Filtres de sécurité
- Remplacer temporairement des paramètres de session
- Modification de valeurs d'invite
- Utilisation de dimensions
- Utilisation de sujets de requête

## Validation de sujets de requête

Vous pouvez valider la définition du sujet de requête sans ouvrir la boîte de dialogue **Définition de sujet de requête**. Cela s'avère utile lorsque

- De nouveaux éléments de requête sont ajoutés à un sujet de requête
- La définition du sujet de requête sous-jacent a été modifiée
- Les paramètres d'une procédure stockée ont été modifiés

## Pourquoi et quand exécuter cette tâche

Des informations sur la validation de sujets requête SAP BW figurent dans une autre rubrique : «Validation de sujets de requête de modèle (SAP BW)», à la page 220.

La commande **Valider l'objet** effectue une vérification exhaustive de tous les sujets de requête et s'assure qu'ils peuvent s'exécuter.

Ce qui se passe dans le processus d'évaluation dépend du type de sujet de requête sélectionné.

Type de sujet de requête	Processus d'évaluation
Sujet de requête de source de données relationnelles	<p>Une requête basée sur les éléments dérivés est envoyée à la source de données relationnelles.</p> <p>La liste des références de source de données est mise à jour.</p> <p>Les attributs physiques, comme le type de données, sont mis à jour selon les besoins.</p>
Sujet de requête de modèle basé sur des métadonnées relationnelles	<p>Une requête basée sur les éléments dérivés est envoyée à la source de données.</p> <p>Le code SQL mis en cache, s'il est disponible, est mis à jour.</p> <p>Les attributs physiques, comme le type de données, sont mis à jour selon les besoins.</p>

Type de sujet de requête	Processus d'évaluation
Sujet de requête de procédure stockée	<p>Une requête basée sur les derniers paramètres de la procédure stockée est envoyée à la source de données.</p> <p>La liste d'éléments de requête dérivés est mise à jour.</p>

Vous pouvez également mettre à jour le sujet de requête («Mise à jour de sujets de requête») s'il provient d'une source de données relationnelles, ou synchroniser la totalité du projet («Synchronisation des projets», à la page 305).

### Procédure

1. Sélectionnez le sujet de requête à évaluer.
2. Cliquez sur **Outils, Valider l'objet**.

### Résultats

Si vous avez modifié la propriété **Agrégat ordinaire** sur **Non pris en charge**, la propriété est réinitialisée lorsque vous évaluez le sujet de requête. Si la propriété est définie sur une autre valeur, elle n'est pas modifiée.

## Mise à jour de sujets de requête

Si vous utilisez une source de données relationnelles, vous pouvez choisir de ne mettre à jour que les sujets de requête au lieu d'exécuter une synchronisation complète du projet. Vous devez effectuer une synchronisation du projet pour synchroniser les modifications effectuées dans une autre source de données.

Le sujet de requête est mis à jour en fonction de la définition de la source de données. Lorsque vous mettez à jour un sujet de requête, de nouvelles métadonnées sont extraites de la source de données et les éléments de requête sont resynchronisés.

Vous pouvez également évaluer le sujet de requête s'il provient d'une source de données relationnelles («Validation de sujets de requête», à la page 96).

Vous ne pouvez pas utiliser la commande **Mettre à jour l'objet** pour les sujets de requête de modèle.

### Procédure

1. Sélectionnez un ou plusieurs sujets de requête.
2. Cliquez sur **Outils, Mettre à jour l'objet**.

**Conseil :** Si vous modifiez l'instruction SQL dans la boîte de dialogue **Définition de sujet de requête**, une opération **Mettre à jour l'objet** est initiée lorsque vous cliquez sur **OK**.

## Conversion de sujets de requête en dimensions

Vous pouvez convertir un sujet de requête en dimension ordinaire ou en dimension de mesure si vous souhaitez utiliser des fonctions associées à des dimensions, telles que la définition de hiérarchies et de niveaux. Un sujet de requête de modèle devient une dimension de modèle.

Vous pouvez convertir des sujets de requête de source de données en dimensions de source de données, mais ces dernières ont des fonctionnalités limitées par rapport aux sujets de requête ou dimensions de modèle. Cessez d'utiliser des dimensions de source de données, qu'il s'agisse de dimensions ordinaires ou de dimensions de mesure. Créez de nouveaux modèles en suivant les meilleures pratiques décrites dans Chapitre 9, «Guide de modélisation des métadonnées», à la page 327 pour utiliser des sujets de requête en tant que base relationnelle du modèle. Définissez des dimensions ordinaires et des dimensions de mesure en tant qu'objets de modèle basés sur des sujets de requête de source de données et/ou des sujets de requête de modèle. Des instructions sur la migration pour les utilisateurs de dimensions de source de données seront fournies dans une édition ultérieure.

Si des déterminants ont été spécifiés pour le sujet de requête, les clés que vous avez indiquées pour chaque déterminant sont utilisées pour générer une hiérarchie avec une clé métier pour chaque niveau. Les déterminants constituent une hiérarchie. Le premier attribut de chaîne associé à chaque déterminant est utilisé pour la légende métier. Si la légende ne correspond pas à l'attribut que vous souhaitez utiliser, vous devez la modifier manuellement.

La conversion d'un sujet de requête en dimension constitue seulement un point de départ. Examinez chaque dimension ainsi créée pour vous assurer qu'elle correspond à vos besoins.

Vous pouvez également convertir une dimension en sujet de requête («Conversion de dimensions ordinaires en sujets de requête», à la page 124).

Vous ne pouvez pas utiliser des déterminants pour créer des hiérarchies distinctes pour la dimension. Vous devez créer les hiérarchies distinctes pour la dimension après avoir converti le sujet de requête.

Vous ne pouvez pas convertir les éléments suivants en dimensions :

- Ensemble de requêtes
- Sujets de requête de procédure stockée
- Sujets de requête SAP BW

### **Procédure**

1. Sélectionnez les sujets de requête à convertir.
2. Cliquez sur **Actions**, **Convertir en dimension ordinaire** ou **Convertir en dimension de mesure**.

## **Conversion de sujets de requête de modèle en sujets de requête de source de données**

Vous pouvez convertir un sujet de requête de modèle en sujet de requête de source de données si vous souhaitez éditer le code SQL.

Ne convertissez pas le sujet de requête de modèle si vous souhaitez qu'il fasse référence à plusieurs sources de données.

### **Procédure**

1. Sélectionnez le sujet de requête de modèle à convertir.
2. Cliquez sur **Actions**, **Convertir en sujet de requête de source de données**.

Cette commande est disponible uniquement si vous avez exécuté la requête et que l'onglet **Informations sur la requête** de la boîte de dialogue **Editer la définition** contient du code SQL.

## Edition du code SQL

SQL est le langage standard pour la création, la mise à jour, et l'interrogation des systèmes de gestion de base de données relationnelle.

Lorsque vous éditez la définition d'un sujet de requête de source de données relationnelles, vous pouvez utiliser

- «Cognos SQL», à la page 102
- «Code SQL natif», à la page 103
- «SQL direct», à la page 104

### Pourquoi et quand exécuter cette tâche

Si vous souhaitez éditer le code SQL d'un sujet de requête de modèle, vous devez le copier à partir de l'onglet **Informations sur la requête** et le coller dans un nouveau sujet de requête de modèle. Vous pouvez également convertir le sujet de requête de modèle en un sujet de requête de source de données. Ne modifiez pas le code SQL si vous souhaitez que le sujet de requête de modèle fasse référence à plusieurs sources de données. Voir «Conversion de sujets de requête de modèle en sujets de requête de source de données», à la page 98.

La modification de l'alias d'une colonne régénère l'élément de requête qui représente cette colonne. Les modifications que vous avez apportées à l'élément de requête ne sont pas conservées car IBM Cognos Framework Manager le considère comme un nouvel élément de requête.

Vous pouvez ajouter des commentaires au code SQL en utilisant `/*` avant le commentaire et `*/` à la fin.

Voici un exemple :

```
select country /* this is a multiline
comment
another line
another line
*/
```

### Procédure

1. Cliquez sur le sujet de requête de source de données à modifier.
2. Cliquez sur **Actions, Editer la définition**.
3. Cliquez sur l'onglet **SQL**, puis faites glisser des objets vers la zone **SQL** ou entrez le code SQL voulu.
4. Cliquez sur **OK**.

## Modification du type de code SQL

Lorsque vous choisissez le type de code SQL dans lequel générer un sujet de requête de source de données, vous devez prendre en compte les facteurs suivants et décider de ce qui est le plus important.

Type	Avantage	Inconvénient
SQL Cognos	SQL Cognos améliore les performances des sujets de requête en supprimant, par exemple, les éléments non utilisés lors de l'exécution de la requête.  SQL fonctionne sur toutes les bases de données prises en charge.	Vous ne pouvez pas entrer un code SQL non standard.
Code SQL natif	Les performances sont optimisées pour tous les sujets de requête liés.  Vous pouvez utiliser un code SQL spécifique pour votre base de données.	Vous ne pouvez pas utiliser un code SQL non pris en charge pour des sous-requêtes.  Le code SQL peut ne pas fonctionner sur un autre type de base de données.
SQL direct	Vous pouvez entrer tout code SQL pris en charge par la base de données.	Framework Manager n'a pas la possibilité d'optimiser automatiquement les performances.  Le code SQL peut ne pas fonctionner sur une autre source de données.

## Prérequis pour le passage en SQL natif

Si vous modifiez le code d'un sujet de requête existant en SQL natif, vous devez d'abord vous assurer que le code SQL respecte les règles qui s'appliquent à la source de données native pour que la requête s'exécute correctement.

Vous devez effectuer les opérations suivantes :

- Editez les noms de table existants.

Cognos SQL utilise une structure en deux parties pour nommer les sujets de requête. Par exemple, [gosales].[ProductLine] signifie que le sujet de requête ProductLine est issu de la base de données gosales. Par conséquent, lorsque vous passez en SQL natif, vous devez vous assurer que tous les noms de table incluent les éléments parent requis par le fournisseur de source de données.

Pour plus d'informations sur les conventions de dénomination, voir «Conventions de dénomination pour les objets d'un projet», à la page 21.

- Assurez-vous que le code SQL est valide pour des sous-requêtes.

IBM Cognos Framework Manager traite les sujets de requête en SQL natif en tant que sous-requêtes. Voici un exemple de sujet de requête Cognos SQL :

```
Select
```

```
P.ProductName, P.Margin From Product P
```

Si vous le changez en SQL natif, Framework Manager génère l'instruction suivante :

```
Select
```

```
oracle_plain.ProductName as Productname,
```

```

oracle_plain.Margin as Margin
From
(GO_SALES1_OR_92_WE...SELECT
P.PRODUCTNAME, P.MARGIN
FROM
PRODUCT P)
)oracle_plain

```

Par conséquent, vous devez vous assurer que le sujet de requête est conforme aux restrictions de base de données supplémentaires imposées aux sous-requêtes, par exemple, ne pas utiliser la clause `With`. SQL direct n'est pas associé aux mêmes restrictions. Cependant, le fait que le code SQL natif soit traité dans le cadre d'une requête plus grande améliore les performances.

Pour tester un code SQL natif à l'aide d'un outil de requête, tel que SQL\*Plus d'Oracle, vous devez placer le code SQL dans la clause `From` d'une instruction `Select`. Par exemple, vous pouvez utiliser la syntaxe suivante dans un outil de requête :

```
Select * from (<Native SQL>) T1
```

- Affectez des alias aux colonnes dérivées.

Affectez des alias aux colonnes dont les valeurs sont calculées. Voici un exemple :

```

SELECT
Length(Country) as LGTH
FROM Country

```

- Placez les noms d'alias entre guillemets.

La modification du type SQL d'un sujet de requête peut modifier la casse des noms d'alias. Lorsque cela se produit, un sujet de requête qui fait référence à l'élément de requête modifié n'est plus valide. Pour éviter que la casse soit modifiée, placez l'alias entre guillemet, par exemple

```

Select
COUNTRY as "test" from COUNTRY

```

- Si un sujet de requête de source de données contient une macro dans la liste de projection (clause `Select`) de l'instruction SQL, spécifiez dans le code SQL un alias qui correspond à la propriété **Nom de colonne** de l'élément de requête.

Une erreur peut se produire car la macro est évaluée à un nom de colonne différent de la valeur la propriété **Nom de colonne** de l'élément de requête correspondante. Le résultat est que le système ne peut pas localiser l'élément dans la liste de projection. Les listes de projection sont statiques.

L'affectation d'un alias garantit que le nom de l'élément reste constant dans la liste de projection lorsque les résultats de l'évaluation de la macro changent.

Par exemple, la requête suivante contient un paramètre de session, `runLocale`, dont la valeur indique la colonne extraite par la requête :

```

Select
#ColumnMap{$runLocale}# as
CountryNameAlias
From
[GoSales].Country

```

Notez que le signe dièse (`#`) est réservé aux macros. Framework Manager supprime tous les éléments figurant en dehors des signes dièse lors de l'exécution de la macro.

## Procédure

1. Cliquez sur le sujet de requête à modifier.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Informations sur la requête**.  
La zone **Résultats du test** est initialement vide tant que vous n'avez pas exécuté la requête.
3. Cliquez sur **Options, Paramètres SQL**.
4. Utilisez la liste **Type SQL** pour modifier le type de code SQL.  
Si vous modifiez le type en SQL natif, utilisez la liste de contrôle ci-dessus pour vous assurer que le code SQL respecte les règles qui s'appliquent à la source de données native.
5. Cliquez sur **OK**.
6. Si vous souhaitez afficher le code SQL, cliquez sur **Tester l'exemple**.
7. Si vous souhaitez afficher la requête réelle, cliquez sur **Requête**.
8. Si vous souhaitez afficher le code xml utilisé par IBM Cognos BI, cliquez sur **Réponse**.
9. Cliquez sur **OK**.

## Cognos SQL

Par défaut, IBM Cognos Framework Manager utilise SQL Cognos pour créer et éditer des sujets de requête. SQL Cognos respecte les normes SQL et fonctionne avec toutes les sources de données tabulaires et relationnelles. Framework Manager génère le code SQL le plus optimisé qui soit. Par conséquent, il est préférable d'utiliser SQL Cognos.

Etant donné que les sujets de requête contenus dans Framework Manager sont semblables aux vues présentes dans des bases de données, le code SQL défini pour chaque sujet de requête doit être conforme aux normes SQL qui s'appliquent à des vues. Par exemple, vous devez affecter des alias à n'importe quelle colonne qui est vide ou dont le nom n'est pas unique. Ce niveau de conformité signifie que SQL Cognos se comporte avec plus de cohérence que le code SQL spécifique à un fournisseur, qui ne respecte pas les normes SQL.

En règle générale, il est préférable d'utiliser SQL Cognos car vous pouvez créer des sujets de requête qui :

- peuvent contenir des métadonnées issues de sources de données multiples ;
- comportent moins de restrictions de base de données ;
- interagissent de manière plus efficace avec des applications IBM Cognos.

### Syntaxe de la norme SQL :

Si la source de données le permet, vous pouvez utiliser la clause `With` avec SQL Cognos. La clause `With` est utilisée pour générer un code SQL plus lisible et permettre à la source de données de générer un plan optimal pour l'extraction des données. La source de données peut plus facilement détecter les cas où les mêmes tables doivent être analysées et peut alors résoudre ces tables en tant que vue ou table temporaire en ligne.

Par défaut, IBM Cognos Framework Manager utilise le constructeur de table commune de la norme SQL lorsque la restriction **Utiliser la clause WITH lors de la génération de SQL** est définie.



Utilisez la clause `With` pour améliorer les performances de la requête si cette dernière est limitée aux fonctionnalités prises en charge par le logiciel de la source de données sous-jacente. Lorsqu'une requête utilise une fonctionnalité qui n'est pas prise en charge par la source de données, l'utilisation de la clause `With` peut provoquer une décomposition supplémentaires de la requête, ce qui peut entraîner une dégradation des performances. Dans ce cas, ne pas utiliser la clause `With` peut générer un ensemble de requêtes mieux adapté pour la source de données sous-jacente.

Voici un exemple de code SQL Cognos utilisant des tables dérivées :

```
SELECT * FROM
(SELECT SNO C1, AVG(QTY) C2, COUNT(*) C3 FROM
SUPPLY
GROUP BY SNO) T1,
(SELECT MAX(QTY) C1 FROM SUPPLY) T2
```

Le code suivant montre comment SQL Cognos transforme l'exemple en clause `With` :

```
WITH T1 AS (SELECT SNO C1, AVG(QTY) C2, COUNT(*)
C3 FROM
SUPPLY GROUP BY SNO),
T2 AS (SELECT MAX(QTY) C1 FROM SUPPLY)
SELECT *FROM T1, T2
```

N'utilisez pas la clause `With` pour un traitement récursif.

Pour plus d'informations sur la clause `With`, voir «Restrictions», à la page 309.

La vérification du type de données et la validation SQL sont constamment améliorées. De ce fait et parce que tous les fournisseurs n'assurent pas une totale conformité avec la norme SQL, les expressions SQL non valides ou ambiguës qui étaient précédemment transmises à la source de données ne le seront plus. Si une expression renvoie un type de données non spécifié par la norme SQL, transmettez l'expression à la source de données en utilisant la syntaxe `{expr}`. Vos utilisateurs doivent utiliser la même technique.

### Code SQL natif

SQL natif correspond au code SQL utilisé par la source de données, par exemple Oracle SQL. Utilisez le code SQL natif pour transmettre l'instruction SQL que vous entrez dans la base de données. IBM Cognos BI peut ajouter des instructions aux données que vous entrez. Vous ne pouvez pas utiliser le code SQL natif dans un sujet de requête qui fait référence à plus d'une source de données dans le projet.

Le code SQL spécifié dans IBM Cognos Framework Manager et traité par la base de données, qu'il soit de type natif ou passe-système, doit être complètement autonome. Il ne doit faire référence à aucun élément extérieur, tel que des invites de base de données, des variables, ou un format natif qui en règle générale doivent être fournis par l'application appelante.

Si vous maîtrisez une version SQL native, vous pouvez l'utiliser pour les sujets de requête qui sont basés sur une source de données unique. En procédant ainsi, vous pouvez utiliser des mots clés qui ne sont pas disponibles dans Cognos SQL et vous pouvez copier et coller du code SQL à partir d'une autre application dans Framework Manager.

Lorsque la requête est générée, Framework Manager combine le code SQL de chaque sujet de requête qui utilise une connexion de source de données indiquée dans une seule requête. Cela permet d'améliorer les performances de la requête. Toutefois, étant donné que le code SQL est généré sous la forme d'une série de sous-requêtes, les requêtes SQL natives doivent se conformer aux restrictions placées par leur fournisseur de base de données sur les tables dérivées.

Voici un exemple de code SQL natif qui renvoie une liste d'employés et de gestionnaires :

```
SELECT
  lpad(' ', (level-1)* 4) ename
EMP_CHART,
  level, empno, ename, job, mgr
FROM
  emp
CONNECT BY PRIOR
  empno = mgr
AND
  deptno not in (20,30)
START WITH
  mgr IS NULL
ORDER BY
  level, job
```

### SQL direct

Utilisez SQL direct lorsque l'instruction SQL entrée n'est pas valide dans une table dérivée. SQL direct vous permet d'utiliser SQL natif sans les restrictions imposées par la source de données aux sous-requêtes. En effet, les sujets de requête SQL directs ne sont pas traités en tant que sous-requêtes. Le code SQL correspondant à chaque sujet de requête est envoyé directement à la source de données dans laquelle les résultats de la requête sont générés.

Les performances sont ralenties car chaque sujet de requête est envoyé à la source de données sous la forme d'une déclaration distincte au lieu d'être optimisé par IBM Cognos Framework Manager. Pour choisir entre SQL natif et SQL direct, vous devez donc décider ce qui est le plus important : les performances ou l'utilisation d'un code SQL qui n'est pas autorisé dans une sous-requête.

En général, il est conseillé de n'utiliser SQL direct que lorsque vous devez créer un sujet de requête contenant une syntaxe qui est spécifique d'une source de données et ne peut pas être utilisée dans une table dérivée, comme dans une clause With ou OrderBy.

Le code SQL spécifié dans Framework Manager et traité par la base de données, qu'il soit natif ou direct, doit être entièrement autonome. Il ne doit pas faire référence à des éléments externes à ce code SQL, par exemple, à des invites de base de données, des variables ou un formatage natif qui sont normalement fournis par l'application appelante.

Voici un exemple de rapport orienté système qui contient la date système :

```
SELECT
  TO_CHAR(SYSDATE, 'DAY, DDTH MONTH YYYY')
FROM
```

SYS.DUAL

Notez que le signe dièse (#) est réservé aux macros et que les noms de colonne doivent être uniques. Framework Manager supprime tous les éléments figurant en dehors des signes dièse lors de l'exécution de la macro.

## Modification du mode de génération du code SQL

Vous pouvez indiquer la manière dont IBM Cognos Framework Manager génère le code SQL qui extrait des données de sources de données relationnelles pour les sujets de requête de source de données ou les sujets de requête de modèle.

Le type de **génération SQL** d'un sujet de requête peut être défini sur **En tant filtre** ou **Minimisée**. La valeur par défaut est **Minimisée**.

Lorsque le type de génération est défini sur **Minimisée**, le code SQL généré contient uniquement l'ensemble de tables et de jointures minimum nécessaire à l'obtention des valeurs pour les éléments de requête sélectionnés.

Lorsque le type de génération est défini sur **En tant filtre**, Framework Manager génère des requêtes contenant l'instruction SQL complète qui a défini le sujet de requête. Utilisez **En tant filtre** lorsque vous souhaitez que la requête soit exécutée en tant que bloc. Le code SQL est traité en tant que vue. Par exemple, vous souhaitez que la requête renvoie le même nombre de lignes à chaque fois qu'elle est exécutée.

L'utilisation de SQL réduit améliore les performances, ce qui se traduit par une exécution nettement plus rapide d'une requête. La génération d'un code SQL réduit est particulièrement avantageux pour les sujets de requête qui représentent des tables de dimension. En utilisant un sujet de requête de modèle unique pour modéliser une dimension, vous pouvez bénéficier de petites requêtes SQL qui s'exécutent beaucoup plus rapidement.

Par exemple, le **type de génération SQL** du sujet de requête suivant est **En tant filtre**. Notez que ce sujet de requête contient une instruction Select imbriquée.

```
select
New_Query_Subject.COUNTRYCODE as COUNTRYCODE,
New_Query_Subject.EUROINUSESINCE as EUROINUSESINCE
from
(select
CONVERSIONRATE.COUNTRYCODE as COUNTRYCODE, COUNTRY.EUROINUSESINCE as EUROINUSESINCE
from
"2 - GOSales1 - OLE-DB".GOSALES1.dbo.CONVERSIONRATE CONVERSIONRATE,
"2 - GOSales1 - OLE-DB".GOSALES1.dbo.COUNTRY COUNTRY
where
(COUNTRY.SALESCOUNTRYCODE = CONVERSIONRATE.COUNTRYCODE)
) New_Query_Subject
```

Si vous modifiez le **type de génération SQL** en **Minimisée**, Framework Manager génère le code SQL simplifié suivant :

```
select
CONVERSIONRATE.COUNTRYCODE as COUNTRYCODE,
COUNTRY.EUROINUSESINCE as EUROINUSESINCE
from
"2 - GOSales1 - OLE-DB".GOSALES1.dbo.CONVERSIONRATE CONVERSIONRATE,
"2 - GOSales1 - OLE-DB".GOSALES1.dbo.COUNTRY COUNTRY
where
(COUNTRY.SALESCOUNTRYCODE = CONVERSIONRATE.COUNTRYCODE)
```

Le code SQL réduit fonctionne mieux lorsque les ensembles de résultats renvoyés de chaque élément de requête sont équivalents. Si des enregistrements dans une colonne ne correspondent pas à des enregistrements d'une autre colonne, le résultat de la requête minimisée génère des lignes supplémentaires. Vous pouvez éviter ce problème en définissant le **type de génération SQL En tant filtre**.

Par exemple, si des types de produits ne sont utilisés par aucun produit et que ces types de produit ont tous une ligne de produits commune, il est signalé une ligne de produit pour laquelle il existe des types de produit, mais pas de produits associés.

### Procédure

1. Cliquez sur le sujet de requête à modifier.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Informations sur la requête**.  
La zone **Résultats du test** est initialement vide tant que vous n'avez pas exécuté la requête.
3. Cliquez sur **Options, Paramètres SQL**.
4. Définissez **Générer SQL** sur **En tant filtre** ou **Minimisée**.
5. Cliquez sur **OK**.
6. Si vous souhaitez afficher le code SQL, cliquez sur **Tester l'exemple**.
7. Si vous souhaitez afficher la requête réelle, cliquez sur **Requête**.
8. Si vous souhaitez afficher le code xml utilisé par IBM Cognos BI, cliquez sur **Réponse**.
9. Cliquez sur **OK**.

### Sujets de requête de modèle et types SQL

Un sujet de requête de modèle qui est basé sur un autre sujet de requête de modèle peut utiliser la logique du sujet de requête parent à la place de sa propre logique. Si le sujet de requête de modèle enfant utilise le type SQL **Réduit**, il n'utilise pas la logique du parent. Si le sujet de requête de modèle enfant utilise le type SQL **Comme une vue**, il utilise la logique du parent.

Par exemple, vous créez un sujet de requête de modèle nommé Produits retournés, qui affiche tous les motifs de retour pour tous les produits. Lorsque vous exécutez la requête Produits retournés, une liste de plus de 700 articles s'affiche. Vous créez ensuite un autre sujet de requête de modèle basé sur le sujet Produits retournés et que vous nommez Motifs de retour. Ce sujet de requête de modèle contient uniquement l'élément de requête Motif de retour. Si le type SQL a pour valeur **Réduit**, le sujet de requête Motifs de retour affiche cinq motifs de retour lorsqu'il est exécuté. Si le type SQL a pour valeur **Comme une vue**, le sujet de requête Motifs de retour utilise la logique du sujet de requête Produits retournés et affiche plus de 700 éléments.

---

## Dimensions

Une dimension représente un grand groupe de données sur un aspect essentiel d'une entreprise, tel que des produits, des dates ou des marchés.

Les types de dimensions que vous pouvez gérer dans IBM Cognos Framework Manager sont les dimensions ordinaires et les dimensions de mesure. Dans SAP BW, les dimensions de mesure sont appelées chiffres clés.

Par exemple, dans un projet d'analyse des ventes, vous incluez ces dimensions :

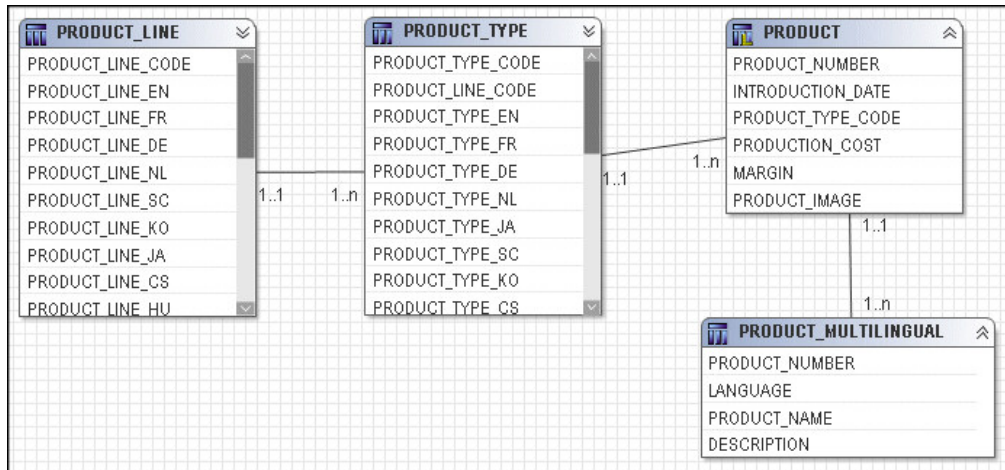
Nom	Saisir	Description
Temps	Dimension ordinaire	Dates de ventes organisées en les années, les trimestres, les mois, les semaines et les jours où les ventes ont été effectuées.
Région	Dimension ordinaire	Emplacements des ventes groupés en régions, pays et villes de vente
Produit	Dimension ordinaire	Détails sur les produits organisés par type de produit, marque, modèle, couleur et emballage.
Client	Dimension ordinaire	Informations client
Ventes	Dimension de mesure	Détails des achats tels que les unités vendues, les revenus et le bénéfice

Vous devez utiliser des dimensions ordinaires et de mesure pour activer une analyse sur votre source de données relationnelles. Dans la plupart des sources de données, les dimensions de mesure sont susceptibles d'être partagées par plusieurs dimensions ordinaires. Les dimensions ordinaires sont souvent appelées "dimensions partagées". Une dimension de mesure et des dimensions ordinaires organisées dans un cluster sont souvent qualifiées de "groupe sous forme de schéma en étoile", mais elles peuvent également être désignées en tant que groupe fonctionnel ou groupe de zones de sujet.

La rubrique suivant peut également vous intéresser : «Sujets de requête et dimensions», à la page 341.

## Sources de données normalisées

Les sources de données normalisées ou en flocons sont souvent dotées de plusieurs tables décrivant un concept professionnel unique. Par exemple, une représentation normalisée du concept Produit peut inclure quatre tables liées par des relations de type 1..n. Chaque catégorie Ligne de produits est constituée d'une ou de plusieurs catégories Types de produit. Chaque catégorie Type de produit est constituée d'une ou de plusieurs catégories Produits. Les produits sont associés à des noms et des descriptions dans plusieurs langues, ils existent donc dans la table de recherche Produit multilingue.



L'une des méthodes de simplification du modèle consiste à créer un sujet de requête de modèle pour chaque concept métier descriptif. Il est possible que vos utilisateurs ne connaissent pas la relation entre les sujets de requête individuels, il est donc utile de les regrouper ; en outre, devoir développer chaque objet de modèle et sélectionner un élément de requête demande plus d'efforts.

L'étape suivante de l'analyse consiste à créer une dimension ordinaire avec un niveau pour chaque sujet de requête.

## Création de dimensions ordinaires

Une dimension ordinaire contient des informations de clé métier et des informations descriptives classées selon une hiérarchie, du niveau de granularité le plus élevé au plus faible. Elle dispose généralement de plusieurs niveaux qui doivent tous avoir une clé et une légende. Si vous n'avez pas de clé unique pour votre niveau, il est conseillé d'en créer une dans un calcul.

Les dimensions ordinaires de modèle sont basées sur des sources de données ou des sujets de requête de modèle déjà définis dans le modèle. Vous devez définir une clé métier et une légende de type de chaîne pour chaque niveau. Lorsque vous vérifiez le modèle, toute absence de clé métier et de légende est détectée. Au lieu de joindre des dimensions ordinaires de modèle aux dimensions de mesure, créez des jointures sur les sujets de requête sous-jacents et créez une relation de portée entre la dimension ordinaire et la dimension de mesure.

Lorsque vous créez des dimensions ordinaires, vous devez comprendre la dimensionnalité des données. Vous devez être en mesure de répondre aux questions suivantes :

- Quels sont les niveaux de dimension ?
- Quels sont l'ordre et la combinaison de niveaux qui constituent des hiérarchies ?
- Quelles sont les relations entre les niveaux ?
- Qu'est-ce qui identifie un niveau de manière unique ?
- Quelles sont les données élémentaires associées à chaque niveau ?
- Disposez-vous de plusieurs niveaux de granularité, par exemple, des données sont-elles enregistrées de façon mensuelle et d'autres de façon quotidienne ?
- Des clés externes sont-elles définies dans la source de données ?

Dans IBM Cognos Framework Manager, vous pouvez indiquer plusieurs hiérarchies pour des dimensions ordinaires. Plusieurs hiérarchies définies pour une dimension ordinaire se comportent comme des vues d'une même requête. Néanmoins, vous ne pouvez utiliser qu'une seule hiérarchie à la fois dans une requête. Par exemple, vous ne pouvez pas utiliser une hiérarchie sur les lignes d'un rapport de type tableau croisé et une autre hiérarchie de la même dimension dans les colonnes. Si vous avez besoin de ces deux hiérarchies dans le même rapport, vous devez créer deux dimensions, une pour chaque hiérarchie. Pour plus d'informations, voir «Modélisation de dimensions avec plusieurs hiérarchies», à la page 358.

Outre la création de dimensions ordinaires, vous pouvez également fusionner des dimensions en une seule dimension ou convertir des sujets de requête en dimensions.

Les requêtes à faits multiples sont activées avec des dimensions conformes.

Vous pouvez utiliser des dimensions de source de données, mais ces dernières ont des fonctionnalités limitées par rapport aux sujets de requête ou aux dimensions de modèle. Cessez d'utiliser des dimensions de source de données, qu'il s'agisse de dimensions ordinaires ou de dimensions de mesure. Créez de nouveaux modèles en suivant les meilleures pratiques décrites dans Chapitre 9, «Guide de modélisation des métadonnées», à la page 327 pour utiliser des sujets de requête en tant que base relationnelle du modèle. Définissez des dimensions ordinaires et des dimensions de mesure en tant qu'objets de modèle basés sur des sujets de requête de source de données et/ou des sujets de requête de modèle. Des instructions sur la migration pour les utilisateurs de dimensions de source de données seront fournies dans une édition ultérieure.

## Procédure de création d'une dimension ordinaire

### Procédure

1. Sélectionnez un espace-noms ou un dossier dans lequel placer la dimension.
2. Cliquez sur **Actions, Créer, Dimension ordinaire**, puis cliquez sur l'onglet **Dimension**.
3. Cliquez sur **Ajouter une hiérarchie**, puis faites glisser un ou plusieurs objets depuis la zone **Éléments disponibles** vers la zone **Hiérarchies**.  
Vous pouvez définir plusieurs hiérarchies pour une dimension. La première hiérarchie est utilisée comme hiérarchie par défaut, ou principale.  
Vous pouvez également créer une hiérarchie secondaire en copiant un niveau. Cliquez sur un niveau et faites-le glisser vers la bordure droite de la dimension. Vous ne pouvez copier un niveau qu'au sein de la même dimension.
4. Cliquez sur **Ajouter un niveau**, puis faites glisser un ou plusieurs objets depuis la zone **Éléments disponibles** vers le nouveau niveau.  
Vous pouvez également créer des copies de niveaux dans la boîte de dialogue **Définition de dimension** ou dans l'onglet **Grille des dimensions**. Cliquez sur le niveau et faites-la glisser vers une autre position dans la hiérarchie. Tous les attributs du niveau sont également copiés. Vous ne pouvez copier un niveau qu'au sein de la même dimension.
5. Si vous souhaitez utiliser un autre élément dans un niveau, faites-le glisser depuis la zone **Éléments disponibles** vers la zone **Sélectionner un niveau dans le contrôle de hiérarchie afin de voir les éléments de requête**.  
Vous êtes invité à indiquer son rôle.  
Par défaut, Framework Manager ajoute le nom à l'espace-noms.


**Astuce :** Pour disposer d'une clé multiparties, par exemple, le prénom plus le nom de famille, créez un attribut qui combine ces éléments, puis indiquez que le nouvel attribut constitue la clé métier.

6. Si vous souhaitez indiquer que les clés de niveaux situées au-dessus du niveau en cours ne sont pas nécessaires pour identifier les membres de ce niveau, sélectionnez l'élément et cochez la case **Niveau unique**. Ceci indique que les valeurs de clé appartenant au niveau doivent être considérées comme étant uniques, quel que soit le contexte. Dans certains cas, le paramètre **Niveau unique** est utilisé par IBM Cognos Software pour optimiser les requêtes SQL.

**Remarque :** La case à cocher **Niveau unique** n'a aucune incidence sur la génération et le traitement des identificateurs de nom unique de membre pour les membres de ce niveau. Tous les noms unique de membre sont entièrement qualifiés.

7. Sélectionnez les tâches supplémentaires à effectuer :
  - Indiquez des rôles («Rôles», à la page 116).
  - Incorporez des calculs en cliquant sur **Ajouter**, puis en définissant l'expression («Création de calculs», à la page 152).  
Pour modifier un calcul qui a été incorporé dans la dimension, dans l'onglet **Grille des dimensions**, cliquez sur **Attributs**, cliquez avec le bouton droit sur l'élément de requête, puis cliquez sur **Modifier l'expression**.
  - Incorporez des filtres («Création de filtres», à la page 155).
  - Indiquez les caractéristiques de tri de niveaux («Tri des membres d'un niveau», à la page 114).
  - Testez la dimension («Test de dimensions», à la page 121).
  - Editez le code SQL et modifiez différentes options («Cognos SQL», à la page 102).
8. Cliquez sur **OK**.
9. Pour modifier la hiérarchie par défaut pour une dimension avec plusieurs hiérarchies, procédez comme suit :
  - Dans la sous-fenêtre **Propriétés**, cliquez sur les points de suspension (...) dans la zone **Hiérarchie par défaut**.
  - Sélectionnez une autre hiérarchie et cliquez sur **OK**.

## Résultats

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour créer une dimension ordinaire. Cliquez sur le bouton de **dimension ordinaire** .

## Hiérarchies pour les dimensions ordinaires

Une hiérarchie est une liste ordonnée de niveaux ou une collection d'éléments. Chaque élément de requête d'une hiérarchie doit porter un nom unique.

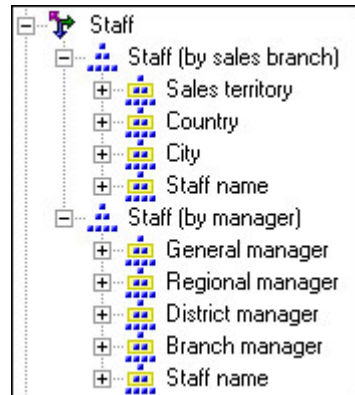
Vous pouvez indiquer des hiérarchies multiples pour des dimensions ordinaires dans IBM Cognos Framework Manager. Plusieurs hiérarchies définies pour une dimension ordinaire se comportent comme des vues d'une même requête. La première hiérarchie est la hiérarchie principale ou par défaut.

Vous ne pouvez utiliser qu'une seule hiérarchie à la fois dans une requête. Par exemple, vous ne pouvez pas utiliser une hiérarchie sur les lignes d'un rapport de type tableau croisé et une autre hiérarchie de la même dimension dans les colonnes. Si vous avez besoin de ces deux hiérarchies dans le même rapport, vous

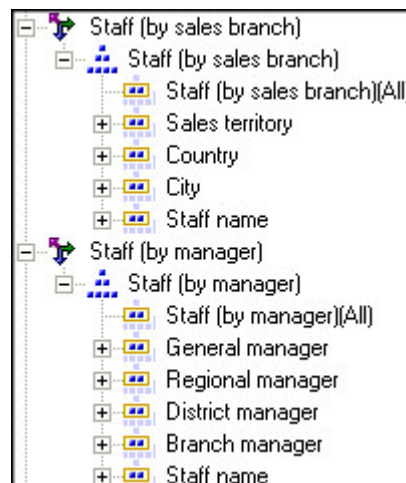


devez créer deux dimensions, une pour chaque hiérarchie. Pour plus d'informations, voir «Modélisation de dimensions avec plusieurs hiérarchies», à la page 358.

Par exemple, l'équipe commerciale (Sales Staff) peut être visualisée par responsable (by manager) ou par succursale (by manager) et peut être modélisée sous la forme d'une dimension unique avec deux hiérarchies.



Si vous avez besoin de ces deux hiérarchies dans la même requête de rapport (par exemple, pour opposer des axes), vous devez créer une dimension ordinaire pour chaque hiérarchie. Par exemple, voici une équipe commerciale avec deux dimensions.



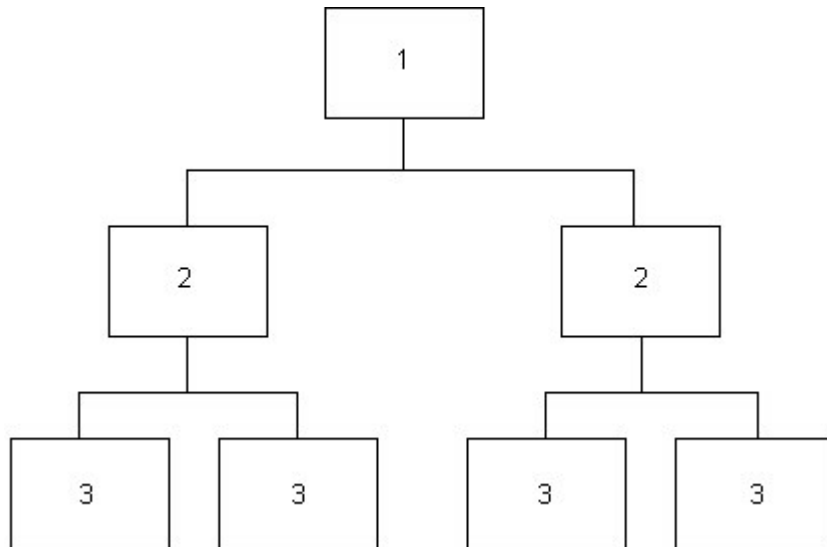
**Astuce :** Pour modifier la hiérarchie par défaut pour une dimension avec plusieurs hiérarchies, dans la sous-fenêtre **Propriétés**, cliquez sur les points de suspension (...) dans la zone **Hiérarchie par défaut** et sélectionnez une autre hiérarchie.

Si une hiérarchie d'une dimension contient un grand nombre de membres, l'exécution d'une requête dans l'un des studios IBM Cognos peut être ralentie car le moteur IBM Cognos génère une requête volumineuse unique pour un cube créé localement. Pour résoudre ce problème, définissez la propriété **Large arborescence des membres** de la sous-fenêtre **Propriétés** sur **true**. Le moteur va alors générer plusieurs petites requêtes pour le cube créé localement.

### Hiérarchie équilibrée :

Chaque chemin présent dans une hiérarchie équilibrée descend à la même profondeur.

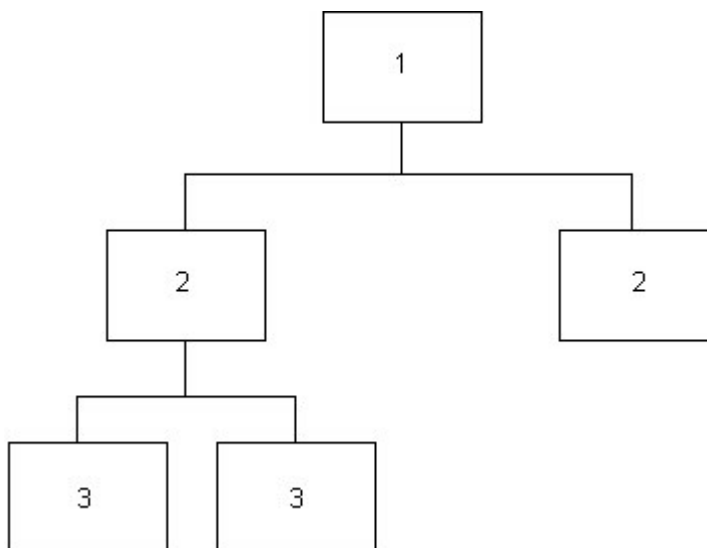
Par exemple, dans le diagramme suivant, le niveau le plus élevé est Ligne de produits (niveau 1), le niveau 2 est Type de produit, puis le niveau 3 est Produits.



### Hiérarchie non équilibrée :

Les branches d'une hiérarchie non équilibrée ont des niveaux de détail différents.

Par exemple, dans le diagramme suivant, le niveau le plus élevé d'une organisation est le PDG (niveau 1) et le niveau 2 est représenté par les vice-présidents et l'adjoint de direction du PDG. L'adjoint de direction n'a pas subordonnés, contrairement aux vice-présidents.



Une hiérarchie non équilibrée peut également être irrégulière. Dans une hiérarchie non équilibrée irrégulière, les niveaux présentent des trous et ont des niveaux de détail différents.

## Hiérarchies irrégulières et de réseau :

Pour les métadonnées relationnelles, mettez à plat les hiérarchies irrégulières et les hiérarchies de réseau dans la source de données.

## Niveaux pour dimensions ordinaires

La définition la plus simple d'un niveau est une clé métier et une légende, chacun d'eux faisant référence à un élément de requête. Une instance (ou ligne) d'un niveau est définie comme membre de ce niveau. Elle est identifiée par un nom unique de membre, lequel contient les valeurs des clés métier du niveau en cours et des niveaux supérieurs. Par exemple, [gosa]es].[Products].[ProductsOrg].[Product]->[All Products].[1].[1].[2] identifie un membre qui se trouve au quatrième niveau, Product, de la hiérarchie ProductsOrg de la dimension [Products] qui est dans l'espace-noms [gosa]es. La légende de ce produit est Gamelle ChefDeCamp, qui est le nom affiché dans l'arborescence de métadonnées et dans le rapport.

Le premier niveau de la hiérarchie prend automatiquement la valeur Tout. Il contient un membre racine unique, lequel représente le niveau supérieur de la hiérarchie. Par exemple, le niveau Tout pour la dimension Temps s'appelle Temps (Tout). Vous ne pouvez pas supprimer ni déplacer le niveau Tout. Vous pouvez modifier son nom, sa description et son infobulle.

Si vous n'indiquez pas les niveaux de la hiérarchie correctement, une agrégation incorrecte peut se produire.

## Noms uniques de membres :

Le nom unique de membre permet d'identifier un membre dans la source de données, tout comme les clés métier qui permettent de rechercher des enregistrements dans une table.

Le nom unique de membre est utilisé dans l'expression d'un élément de données de membre inclus dans un rapport et dans une référence à des membres dans des filtres et dans des expressions. En outre, il est utilisé lors des opérations d'accès aux détails entre les sources de données OLAP. Les clés de membre figurant dans le nom unique de membre des différentes sources de données OLAP doivent correspondre.

Si un nom unique de membre est modifié, les membres qui sont directement référencés dans des expressions, des filtres ou des rapports deviennent introuvables car le nom unique de membre est contenu dans la spécification du rapport. Les raisons pour lesquelles un nom unique de membre peut être modifié sont les suivantes :

- Les modifications apportées aux structures de hiérarchie et de niveau peuvent entraîner la modification du nom unique de niveau.
- Les valeurs de clé métier ont été modifiées et cela a eu pour conséquence de modifier le chemin de clé de membre.
- L'application a changé au cours de la conception ou avec le temps.
- Le cube comporte des codes de catégorie dont l'unicité est imprévisible.
- L'environnement de production comporte plus de membres que l'environnement de test.
- Le membre ne figure plus dans la source de données.

Pour éviter ces problèmes, utilisez les meilleures pratiques suivantes :

- Utilisez des codes et des clés uniques au sein d'une dimension pour les clés des membres.
- Utilisez des valeurs conformes uniques pour les dimensions similaires entre les environnements cible et source lorsque vous activez l'accès aux détails.
- Lorsque vous utilisez un modèle relationnel modélisé de façon dimensionnelle pour l'accès aux détails, prenez soin de vérifier que la clé métier racine est conforme aux membres racine des autres sources de données.
- Assurez-vous que les clés métier et la structure des métadonnées dimensionnelles sont identiques dans les environnements de production et de test.
- Ne modifiez pas les clés métier dans IBM Cognos Framework Manager après le passage en production.
- Procédez à la résolution des clés non uniques dans une dimension de la source de données. N'utilisez pas de tildes dans les codes de catégorie.
- Si vous disposez d'un cube Transformer créé avec des valeurs source non uniques, n'utilisez pas la fonction Supprimer les catégories de Transformer car celle-ci modifiera probablement les codes de catégorie. Conservez une copie de sauvegarde de votre fichier MDL.

#### Clés pour niveaux :

Une clé est un élément de requête qui identifie de manière unique les membres d'un niveau. Par exemple, l'élément Numéro de produit identifie de manière unique un produit alors que les éléments Ville, Etat et Pays sont tous nécessaires pour identifier une ville de manière unique. La clé peut ou non être contenue dans un niveau. Les clés externes sont utilisées pour relier la dimension de mesure à ses dimensions ordinaires.

Chaque niveau nécessite un élément qui est défini en tant que clé.

Si une dimension de modèle contient un élément de requête dont le type de données est BLOB, créez un sujet de requête avec des déterminants, puis créez une dimension de modèle qui est basée sur le sujet de requête de modèle.

## Tri des membres d'un niveau

Pour les métadonnées relationnelles modélisées sous forme de dimension, vous pouvez spécifier des caractéristiques de tri sur une dimension. Vous pouvez également définir un tri sur des niveaux individuels au sein de la dimension. Lorsque vous trie des niveaux individuels au sein d'une dimension, vous pouvez définir un ordre approprié pour les données transmises au modèle. L'ordre dans lequel les données sont transmises au modèle peut être particulièrement important lorsque de la génération de rapports de données sur des périodes de date relative.

L'ordre de tri par défaut est alphabétique en ordre croissant, selon le niveau de légende. Lorsque vous indiquez un ordre de tri, les données sont triées sur une autre colonne, par exemple, la clé métier.

### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, sélectionnez une dimension ou un niveau.
2. Cliquez sur **Actions, Editer la définition**.
3. Cliquez sur l'onglet **Tri des membres**.
4. Sélectionnez l'option de tri à appliquer.

Options de tri	Description
Métadonnées (telles qu'elles sont présentées dans l'arborescence des membres)	<p>Utilisée uniquement pour indiquer la façon dont les membres sont triés lorsqu'ils sont affichés dans l'arborescence de métadonnées.</p> <p>Le tri de l'arborescence de métadonnées ne peut pas être modifié par les auteurs de rapports.</p>
Données - Uniquement en tant que tri de rapport par défaut	<p>Utilisée en tant que tri des données par défaut pour les membres d'un rapport.</p> <p>Si aucun tri de données n'est spécifié, les données sont extraites dans l'ordre dans lequel elles ont été entrées dans la base de données.</p> <p>Les auteurs de rapports peuvent remplacer cette valeur et appliquer un tri différent aux membres affichés dans un rapport.</p>
Données - Toujours (compatible OLAP)	<p>Utilisée pour fournir des fonctions relatives de membre (MRF) avec un ordre cohérent des membres. Ce paramètre doit être utilisé uniquement si des fonctions relatives de membre sont requises. Sinon, le tri se traduit par une surcharge qui n'est pas nécessaire. Les auteurs de rapports ne peuvent pas modifier l'ordre de membres transmis aux fonctions relatives de membre. Toutefois, ils peuvent appliquer un tri différent aux membres affichés dans le rapport.</p> <p>Si aucun tri n'est indiqué et que des fonctions relatives de membre sont utilisées, l'auteur du rapport reçoit une erreur lorsque des fonctions relatives de membre sont traitées.</p> <p>Lorsque cette option est définie, les membres du niveau sont également triés dans l'arborescence de métadonnées même si l'option Métadonnées n'est pas sélectionnée. Le tri de l'arborescence de métadonnées ne peut pas être modifié par les auteurs de rapports.</p>

#### Astuces

- Pour appliquer l'ordre de tri par défaut à tous les niveaux enfant au sein des niveaux de la dimension ne possédant pas d'option de tri définie, cliquez sur **Détecter**.
  - Pour supprimer les options de tri de tous les niveaux enfant au sein des niveaux de la dimension, cliquez sur **Effacer tout**.
5. Dans la zone **Sélectionner un niveau à attribuer aux propriétés de tri**, cliquez sur un niveau.
  6. Dans la zone **Éléments de données disponibles**, cliquez sur un élément à trier, puis cliquez sur la flèche de droite pour ajouter l'élément à la zone **Propriétés du tri de niveau**. Utilisez les flèches vers le haut ou vers le bas pour modifier l'ordre des éléments.

**Astuce :** Pour modifier l'ordre de tri en croissant ou décroissant, cliquez sur **Ordre de tri**.

7. Pour spécifier la façon dont les valeurs nulles sont triées dans les rapports, cliquez dans la zone **Nulles** en regard de l'élément, puis cliquez sur **Premier**, **Dernier** ou **Non spécifié**.

Premier place les valeurs nulles au début et Dernier les place en bas. Non spécifié utilise le paramètre défini dans la source de données.

8. Cliquez sur **OK**.

## Rôles

Les rôles définissent ce qui s'affiche dans l'arborescence de membres dans les studios IBM Cognos. Utilisez des rôles pour organiser et gérer les métadonnées, et déterminer comment présenter les données aux utilisateurs.

Vous pouvez également créer des expressions qui font référence à des rôles au lieu de créer des éléments de requête. Vous devez utiliser la fonction `roleValue` pour faire référence à un rôle particulier. Par exemple, vous souhaitez exécuter une requête sur un rôle spécifique dans une hiérarchie, mais l'élément de requête assumant ce rôle est différent à chaque niveau de la hiérarchie. Une requête unique peut s'appliquer aux différents éléments de requête à chaque niveau. Vous pouvez également utiliser la fonction `roleValue` lorsque vous connaissez le rôle mais pas l'élément de requête sous-jacente.

Vous pouvez affecter plusieurs rôles à un élément de requête, mais le rôle même ne peut pas être attribué à des éléments de requête différents dans le même niveau.

Des rôles par défaut sont prédéfinis pour toutes les hiérarchies parent-enfant et pour tous les niveaux des hiérarchies basées sur des niveaux. La plupart des rôles ne sont pas visibles dans les studios IBM Cognos.

Les rôles qui sont réservés par IBM Cognos BI commencent par un trait de soulignement. Le nom d'un rôle personnalisé ne peut donc pas commencer par un trait de soulignement.

### Rôles par défaut

Les rôles par défaut sont les suivants :

- `_businessKey`

Représente la clé pour le niveau. Le niveau peut être défini comme unique si la clé métier du niveau est suffisante pour identifier chaque ensemble de données pour un niveau.

Le rôle `_businessKey` ne peut être affecté qu'à un seul attribut dans un niveau.

La propriété **Clé métier racine** affiche la valeur de la clé métier pour le membre racine. Le membre racine est un niveau artificiel créé pour les modèles de données relationnelles dimensionnelles. Pour permettre l'accès au détail sur des dimensions conformes, vous devez définir la propriété **Clé métier racine**.

- `_memberCaption`

Présente la légende pour un membre qui s'affichera dans les studios IBM Cognos.

Le rôle `_memberCaption` est nécessaire pour tirer parti des fonctions membres et pour pouvoir déplacer des niveaux par glisser-déposer dans les studios IBM Cognos.

Assurez-vous que le type de données est défini sur chaîne pour l'élément auquel sera affecté le rôle `_memberCaption`.

- `_memberDescription`

Renvoie la description d'un membre au sein d'une dimension.

## Rôles personnalisés

Par défaut, les attributs sont inclus sans rôle. Vous pouvez affecter des attributs à des rôles existants ou créer des rôles personnalisés. Chaque rôle que vous créez doit porter un nom unique.

Vous pouvez traduire les rôles personnalisés dans le modèle.

## Spécification de rôles

Les rôles définissent ce qui s'affiche dans l'arborescence de membres dans les studios IBM Cognos. Utilisez des rôles pour organiser et gérer les métadonnées, et déterminer comment présenter les données aux utilisateurs.

### Procédure

1. Cliquez sur la dimension pour laquelle vous souhaitez définir des rôles.
2. Dans le menu **Actions**, cliquez sur **Editer la définition**.
3. Cliquez sur l'onglet **Dimension**.
4. Dans la zone **Hierarchies**, cliquez sur le niveau souhaité.
5. Dans la zone **Sélectionner un niveau dans le contrôle de hiérarchie afin de voir les éléments de requête**, cliquez sur un élément de requête.
6. Sous **Rôle**, cliquez sur les points de suspension (...).
7. Effectuez l'une des opérations suivantes :
  - Pour utiliser un rôle défini par Framework Manager, cliquez sur l'onglet **Rôles par défaut** et sélectionnez un rôle.
  - Pour créer un rôle, cliquez sur l'onglet **Rôles personnalisés**, puis sur **Ajouter**.
8. Cliquez sur **Fermer**.
9. Cliquez sur **OK**.

### Résultats

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour définir des rôles. Cliquez sur **Attributs**, cliquez avec le bouton droit de la souris sur l'élément de requête et cliquez sur **Editer les rôles**.

## Création de dimensions de mesure

Seules les mesures sont visibles dans la dimension de mesure de modèle. Les éléments de requête, tels que les clés, sont masqués.

Une dimension de mesure est une série de faits. Vous pouvez créer une dimension de mesure pour un ou plusieurs sujets de requête entre lesquels il existe une relation valide.

Les dimensions de mesure de modèle doivent être exclusivement composées d'éléments quantitatifs. Par nature, il n'existe pas de clés sur lesquelles appliquer la jointure, il est donc impossible de créer des jointures vers des dimensions de mesure de modèle. Au lieu de joindre des dimensions de mesure de modèle aux

dimensions ordinaires, créez des jointures au niveau des sujets de requête sous-jacents. Ensuite, créez manuellement une relation de portée entre eux ou bien détectez cette portée si les deux dimensions se trouvent dans le même espace-noms.

Vous pouvez utiliser des dimensions de source de données, mais ces dernières ont des fonctionnalités limitées par rapport aux sujets de requête ou aux dimensions de modèle. Cessez d'utiliser des dimensions de source de données, qu'il s'agisse de dimensions ordinaires ou de dimensions de mesure. Créez de nouveaux modèles en suivant les meilleures pratiques décrites dans Chapitre 9, «Guide de modélisation des métadonnées», à la page 327 pour utiliser des sujets de requête en tant que base relationnelle du modèle. Définissez des dimensions ordinaires et des dimensions de mesure en tant qu'objets de modèle basés sur des sujets de requête de source de données et/ou des sujets de requête de modèle. Des instructions sur la migration pour les utilisateurs de dimensions de source de données seront fournies dans une édition ultérieure.

Vous pouvez ajouter de la valeur en intégrant des calculs basés sur des règles métier existants, tels que la marge bénéficiaire.

Vous pouvez changer l'ordre des mesures, des éléments de requête et des calculs.

Si la dimension de mesure contient un dossier, vous pouvez changer l'ordre des éléments uniquement dans l'**afficheur de projet**.

Vous ne pouvez pas définir des hiérarchies ou des niveaux pour une dimension de mesure.

## Procédure

1. Cliquez sur l'espace-noms dans lequel vous souhaitez placer la dimension de mesure.
2. Cliquez sur **Actions, Créer, Dimension de mesure**.
3. Cliquez sur l'onglet **Dimension de mesure**.
4. Faites glisser des mesures depuis la zone **Objets de modèle** vers la zone **Mesures**.
5. Effectuez les actions voulues.

Objectif	Action
Intégrer un calcul	Cliquez sur <b>Ajouter</b> . Vous pouvez également cliquer avec le bouton droit de la souris sur une mesure et cliquer sur <b>Ajouter</b> ou <b>Editer</b> .
Intégrer un filtre	Cliquez sur l'onglet <b>Filtres</b> .
Tester la dimension de mesure	Cliquez sur l'onglet <b>Test</b> ou <b>Informations sur la requête</b> .
Convertir une mesure en élément de requête	Cliquez avec le bouton droit de la souris sur la mesure et cliquez sur <b>Convertir en élément de requête</b> .



**Remarque :** Si vous testez la dimension de mesure en utilisant l'onglet **Informations sur la requête**, IBM Cognos BI valide la dimension de mesure. Si vous testez la dimension de mesure à l'aide de l'onglet **Test**, IBM Cognos BI exécute la dimension de mesure. Le code SQL pour la validation est légèrement différent de celui pour l'exécution. Pour générer un code SQL définitif pour la dimension de mesure, utilisez l'onglet **Test**.

6. Cliquez sur **OK**.

## Résultats

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour créer une dimension de mesure. Cliquez sur le bouton de **dimension de mesure** .

Vous pouvez être amené à consulter les rubriques associées suivantes :

- «Requêtes à granularité multiple et à faits multiples», à la page 335
- «Création de calculs», à la page 152
- «Création de filtres», à la page 155
- «Test de dimensions», à la page 121
- «Modification du mode d'agrégation des éléments de requête», à la page 136

## Conversion de mesures en éléments de requête

Si vous avez créé une dimension de mesure et que vous souhaitez la joindre à des dimensions ordinaires, vous devez créer des jointures. Les jointures ont besoin de clés et les clés sont des éléments de requête, pas des mesures. La mesure que vous souhaitez utiliser comme clé doit être convertie en élément de requête.

Vous pouvez également convertir un élément de requête en mesure («Conversion d'éléments de requête en mesures», à la page 151).

### Procédure

1. Cliquez deux fois sur la dimension de mesure contenant la mesure.
2. Cliquez sur l'onglet **Dimension de mesure**.
3. Cliquez avec le bouton droit de la souris sur la mesure, puis cliquez sur **Convertir en élément de requête, OK**.

## Relations de portée

Les relations de portée sont nécessaires pour définir les dimensions et les mesures qui sont utilisées ensemble pour des modèles relationnels modélisés de façon dimensionnelle.

Une relation de portée est automatiquement générée lorsque vous faites glisser une dimension dans la grille des dimensions ou lorsque vous déplacez un sujet de requête dans l'espace-noms de la dimension et que vous le convertissez en une dimension ordinaire.

**Remarque :** les raccourcis vers les relations de portée ne sont pas pris en charge.

Les relations de portée sont établies uniquement entre des dimensions de mesure et des dimensions ordinaires pour définir le niveau auquel les mesures sont disponibles à des fins de génération de rapports. Elles sont différentes des jointures et n'ont aucune incidence sur la clause *Where*. Dans une relation de portée, aucune condition ni critère ne sont définis sur la façon de former une requête. Elle indique

uniquement si une dimension peut faire l'objet de requêtes avec un fait spécifique. En l'absence de relation de portée, une erreur se produit lors de l'exécution.




Si vous définissez la relation de portée pour la dimension de mesure, les mêmes paramètres s'appliquent à toutes les mesures de la dimension de mesure. Si les données font l'objet de rapports à un niveau différent des mesures dans la dimension de mesure, vous pouvez définir la portée d'une mesure. Vous pouvez définir le niveau le plus bas à partir duquel les données peuvent faire l'objet d'un rapport.

Lorsque vous créez une dimension de mesure, IBM Cognos Framework Manager crée une relation de portée entre la dimension de mesure et chaque dimension ordinaire existante. Framework Manager recherche les chemins de jointure entre la dimension de mesure et les dimensions ordinaires, en commençant par le niveau de détail le plus bas. Si de nombreux chemins de jointure sont disponibles, la relation de portée créée par Framework Manager peut ne pas être celle prévue. Dans ce cas, vous devez éditer la relation de portée.


## Définition de relations de portée

Les relations de portée sont établies uniquement entre des dimensions de mesure et des dimensions ordinaires pour définir le niveau auquel les mesures sont disponibles à des fins de génération de rapports.

### Procédure

1. Cliquez sur l'onglet **Grille des dimensions**.  
**Astuce :** Pour afficher les relations de portée mises en évidence avec une couleur d'arrière-plan, cliquez sur le bouton **Afficher la portée** .
2. Cliquez sur une ou plusieurs dimensions de mesure.
3. Cliquez sur le niveau de la dimension sur laquelle vous souhaitez définir la portée.  
**Astuce :** Si vous souhaitez que Framework Manager définisse la relation de portée, sélectionnez la dimension de mesure et la dimension ordinaire, puis cliquez sur le bouton **Déterminer la portée** .
4. Cliquez sur le bouton **Définir la portée** .

### Résultats

Si vous souhaitez supprimer la portée, sélectionnez la hiérarchie ou la dimension, puis cliquez sur le bouton **Retirer la portée** .

Si vous sélectionnez une hiérarchie, vous pouvez supprimer la portée d'une hiérarchie spécifique sans affecter la portée définie dans d'autres hiérarchies de la dimension.

Si vous sélectionnez la dimension, la portée de toutes les hiérarchies est supprimée. La relation de portée entre la dimension de mesure et la dimension ordinaire est également supprimée.

## Création de dimensions ordinaires basées sur des objets existants

Vous pouvez créer une dimension ordinaire en fusionnant des objets existants. Ces objets peuvent être des dimensions, des sujets de requête ou des éléments de requête.

### Procédure

1. Sélectionnez les objets que vous souhaitez inclure dans une dimension.
2. Cliquez sur **Actions, Fusionner dans la nouvelle dimension ordinaire**.

## Affichage d'objets associés

Vous pouvez explorer une représentation visuelle des objets qui sont connectés au sujet de requête ou à la dimension que vous sélectionnez dans l'**afficheur de projet**. L'explorateur de contexte **Context Explorer** affiche les objets auxquels l'objet sélectionné est connecté. Vous pouvez également sélectionner un objet connecté et voir ses références.




Vous pouvez masquer un objet dans **Context Explorer**. Vous pouvez aussi modifier la présentation, ajuster tous les objets dans **Context Explorer**, effectuer un zoom avant ou arrière, imprimer, afficher un aperçu des diagrammes avant impression et modifier la mise en page.

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour explorer les dimensions.

### Pourquoi et quand exécuter cette tâche

#### Procédure

1. Sélectionnez un ou plusieurs objets à explorer.
2. Dans le menu **Outils**, cliquez sur **Lancer Context Explorer**.
3. Pour afficher les objets connectés, cliquez sur un ou plusieurs objets, puis sur le bouton approprié.

Objectif	Bouton
Afficher les objets qui sont associés à l'objet sélectionné.	
Afficher les références immédiates pour les objets.	
Afficher toutes les références pour les objets.	

4. Si vous souhaitez afficher des détails sur un objet, par exemple, ses relations et ses éléments de requête, cliquez avec le bouton droit de la souris sur l'objet, cliquez sur **Parcourir le diagramme, Paramètres de diagramme**, puis sélectionnez les détails voulus.

## Test de dimensions

Le test d'une dimension ordinaire renvoie les attributs associés à la hiérarchie définie comme hiérarchie par défaut.

## Pourquoi et quand exécuter cette tâche

### Test d'objets

Vous pouvez voir les résultats renvoyés par un objet en le testant. Vous pouvez tester un objet lorsque vous le créez ou ultérieurement. Les objets que vous pouvez tester sont les dimensions, les sujets de requête, les ensembles de requêtes, les hiérarchies, les niveaux, les calculs, et les éléments de requête.

Vous pouvez visualiser les données qui seront affichées dans un rapport spécifique avant de publier un pack en sélectionnant et testant les objets qui apparaîtront dans le rapport. Il est ainsi plus facile de déboguer un modèle et de vérifier que celui-ci répond aux exigences des rapports, car vous n'avez pas besoin de créer et de publier d'abord des packs.

Lorsque vous testez un objet, IBM Cognos Framework Manager renvoie des exemples de données. Aucun formatage n'est appliqué aux exemples de données. Si vous devez tester un formatage, vous devez publier le pack et afficher les objets dans les studios IBM Cognos.

Vous pouvez voir différents résultats en fonction de ce que vous testez. Par exemple, si vous utilisez l'éditeur d'expression pour tester un calcul qui est intégré dans un sujet de requête, Framework Manager teste uniquement l'expression, pas l'élément. Le paramètre d'agrégation pour l'élément de requête n'est donc pas appliqué au test. Le test de la totalité du sujet de requête, qui inclut le calcul, donne un résultat différent car le paramètre d'agrégation est appliqué. Par exemple, si le paramètre d'agrégation a pour valeur Récapituler, vous verrez moins de lignes dans le test.

Lorsque vous testez une dimension de mesure, SQL utilise des agrégats, pas les mesures.

Si vous testez un segment enfant d'un modèle segmenté, une erreur peut s'afficher si un objet que vous testez fait référence à un objet d'un autre segment enfant et que l'objet référencé n'est pas disponible pour le projet en cours. Vérifiez que le modèle parent contient tous les objets et que ce message d'erreur ne s'affiche pas lorsque vous testez le modèle parent.

Des paramètres de restriction peuvent affecter les résultats de test. Pour plus d'informations, voir «Restrictions», à la page 309.

Vous pouvez modifier les paramètres de test existants pour personnaliser les résultats affichés par le test. Par exemple, en plus d'autres paramètres, vous pouvez contrôler le nombre de lignes renvoyées.

## Étapes lors de la création ou de la modification de l'objet

### Procédure

1. Sélectionnez l'objet à tester.
2. Cliquez sur **Actions**, **Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.

La zone **Résultats du test** est initialement vide tant que vous n'avez pas exécuté la requête.

Tous les ensembles de résultats qui contiennent des objets BLOB (Binary Large Object) sont affichés en tant que [blob].

3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Si vous souhaitez ajouter un nombre de lignes, cliquez sur **Nombre total de lignes**.
5. Si vous souhaitez appliquer la propriété **Agrégat ordinaire** de l'élément de requête ou la propriété **Agrégat des règles** d'une mesure de semi-additive qui est référencée dans l'expression, cochez la case **Total automatique**.  
Si vous désactivez cette case, une ligne est renvoyée pour chaque ligne de l'ensemble de résultats de la requête.
6. Si vous souhaitez obtenir plus d'informations sur les résultats de requête, cliquez sur l'onglet **Informations sur la requête**.
7. Cliquez sur **OK**.

### Procédure de visualisation des données qui seront affichées dans un rapport spécifique

#### Procédure

1. Sélectionnez les objets qui s'afficheront dans le rapport.
2. Cliquez sur **Outils, Test**.
3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Pour afficher des détails sur tout problème trouvé, cliquez sur l'onglet **Informations sur la requête**.

Si vous ne voyez pas les résultats de la requête dans la fenêtre de test, il se peut que les données de votre source de données dépassent la valeur de l'une des restrictions. La requête s'arrête à la limite spécifiée, mais la fenêtre des résultats de test ne contient aucune donnée. **Astuce** : Définissez chaque restriction sur zéro.

#### Modification des paramètres de test

Vous pouvez personnaliser les tests en modifiant les paramètres de test.

#### Procédure

1. Sélectionnez l'objet de votre choix.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.
3. Cliquez sur **Options, Tester les paramètres**.
4. Sélectionnez les options de votre choix.

Objectif	Action	Persistance
Extraire toutes les données et afficher un nombre spécifique de lignes	Cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes.  Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.	Ce paramètre s'applique à toutes les dimensions, tous les sujets de requête et tous les ensembles de requêtes du modèle.  Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.

Objectif	Action	Persistance
Indiquer le niveau de détail	Faites glisser le curseur de <b>Niveau de détail affiché dans les informations de résultat</b> pour représenter la quantité d'informations dont vous avez besoin.	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.
Remplacer temporairement des paramètres de session	Dans la zone <b>Paramètres de session</b> , cliquez sur <b>Définir</b> .  La boîte de dialogue <b>Paramètres de session</b> s'affiche.	Ces valeurs de remplacement ne sont pas enregistrées avec le modèle. Ce paramètre est disponible uniquement pour la session en cours.
Appliquer les filtres de mode de conception appropriés	Cochez la case <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b> .  Ce paramètre s'applique à tous les filtres pertinents dont l'utilisation est définie sur le mode de conception dans une autre dimension, un autre sujet de requête ou un autre ensemble de requêtes.	Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.
Appliquer un filtre de sécurité	Dans la zone <b>Filtres de sécurité</b> , cliquez sur <b>Editer</b> .	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.
Modifier les valeurs d'invites	Dans la zone <b>Valeurs d'invite en cours</b> , cliquez sur <b>Invites</b> .  La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.	Les valeurs d'invite ne sont pas enregistrées avec le modèle.  Ce paramètre est disponible uniquement pour la session en cours.

5. Cliquez sur le bouton **OK** à deux reprises.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Utilisation de dimensions : «Dimensions», à la page 106
- Utilisation de sujets de requête :«Sujets de requête», à la page 76

## Conversion de dimensions ordinaires en sujets de requête

Vous pouvez convertir une dimension ordinaire en sujet de requête de modèle ou en sujet de requête de source de données.

Vous pouvez également convertir un sujet de requête en dimension. Voir «Conversion de sujets de requête en dimensions», à la page 97.

Si une dimension comprend plusieurs hiérarchies, seule la hiérarchie par défaut est incluse lorsque vous convertissez la dimension en un sujet de requête.

### **Procédure**

1. Cliquez sur la dimension ordinaire.
2. Cliquez sur **Actions, Convertir en sujet de requête**.

---

## **Métadonnées multilingues**

Pour les modèles qui sont publiés dans plusieurs langues, vous pouvez afficher et modifier les objets de modèle dans les différentes langues.

Vous devez gérer le support multilingue dans la vue d'importation pour un certain nombre de raisons. Vous pouvez réduire le nombre d'éléments de requête contenus dans chaque dimension et sujet de requête. Le modèle est plus facilement gérable s'il comporte moins de dimensions, de sujets de requête et d'éléments de requête. Vous pouvez simplifier la maintenance en effectuant tout le travail multilingue à un seul endroit plutôt que dans diverses vues métier. Cela garantit une certaine cohérence car les langues sont correctement configurées pour tous les modélisateurs à utiliser. Cela est particulièrement important pour les modèles segmentés.

### **Prise en charge des métadonnées multilingues**

Plusieurs étapes sont requises pour permettre la prise en charge des métadonnées multilingues.

#### **Procédure**

1. Importez des métadonnées à partir de sources de données multilingues. Voir «Modélisation avec des sources de données multilingues», à la page 127.
2. Définissez les langues prises en charge par le modèle. Voir «Ajout d'une langue à un projet», à la page 129.
3. Définissez une ou plusieurs tables de mappage des paramètres qui traduisent les paramètres régionaux utilisés lorsque le rapport est exécuté dans les valeurs de langue de la source de données. Voir «Création de tables de mappage des paramètres», à la page 160.
4. Utilisez une macro pour remplacer dynamiquement des valeurs de langue à partir de la table de correspondance des langues à l'aide du paramètre de session `runLocale` comme clé. Voir «Création d'invites avec des macros de requête», à la page 165.
5. Exportez des propriétés multilingues dans des tables de traduction utilisées par les traducteurs pour entrer le texte approprié pour chaque langue. Voir «Exportation des tables de traduction», à la page 130.
6. Importez la table qui contient les valeurs de propriété traduites. Voir «Importation des tables de traduction», à la page 130.
7. Publiez les métadonnées dans les langues spécifiées. Voir Chapitre 7, «Publication de packs», à la page 255.

## Résultats

Pour des informations sur la façon d'activer la modélisation multilingue, voir «Exemple - Création de projets multilingues pour des métadonnées relationnelles», à la page 131.

## Configuration d'un environnement de génération de rapports multilingue

Vous pouvez créer des rapports qui affichent les données dans plusieurs langues et qui utilisent différents paramètres régionaux. Vous pouvez ainsi créer un rapport unique qui peut être consulté par des utilisateurs de rapports du monde entier.

Les exemples de bases de données fournis avec IBM Cognos comprennent une sélection de zones de texte, tels que des noms et des descriptions, dans plus de 25 langues, destinée à présenter un environnement de génération de rapports multilingue. Pour en savoir davantage sur la manière dont les données sont stockées dans les exemples de bases de données et la façon dont ces derniers sont configurés pour utiliser des données multilingues, reportez-vous au *Guide d'administration et de sécurité d'IBM Cognos Business Intelligence*.

Voici comment se déroule la création d'un environnement de génération de rapports multilingue :

- Utilisation de métadonnées multilingues  
L'administrateur de source de données peut stocker des données multilingues dans des tables, des lignes ou des colonnes individuelles.  
Pour en savoir davantage sur la configuration de votre base de données à des fins de génération de rapports multilingues, reportez-vous au *Guide d'administration et de sécurité d'IBM Cognos Business Intelligence*.
- Création d'un modèle multilingue  
Les modélisateurs utilisent Framework Manager pour ajouter des métadonnées multilingues au modèle à partir de n'importe quel type de source de données, à l'exception de OLAP. Pour ce faire, ils définissent les langues prises en charge par le modèle, traduisent les chaînes de texte dans le modèle pour des éléments tels que les descriptions et les noms d'objet et définissent les langues exportées dans chaque pack. Si la source de données contient des données multilingues, les modélisateurs peuvent définir des requêtes permettant d'extraire les données dans la langue par défaut de l'utilisateur du rapport.  
Pour en savoir davantage, reportez-vous au *Guide d'utilisation d'IBM Cognos Framework Manager*.
- Création de mappages multilingues  
Les administrateurs et les modélisateurs font appel à un utilitaire du système d'exploitation Microsoft Windows appelé Map Manager pour importer des cartes et mettre à jour les libellés associés dans IBM Cognos Report Studio. Pour les éléments de cartes, tels que les noms de pays ou de région et les noms de ville, les administrateurs et les modélisateurs peuvent définir d'autres noms afin de fournir des versions multilingues du texte apparaissant sur la carte.  
Pour plus d'informations, reportez-vous au *Guide d'installation et d'utilisation d'IBM Cognos Map Manager*.
- Création d'un rapport multilingue  
Le créateur du rapport utilise Report Studio pour créer un rapport qui peut être affiché dans plusieurs langues. Vous pouvez par exemple indiquer qu'une partie du texte (le titre par exemple) doit apparaître en allemand lorsque le rapport est



ouvert par un utilisateur allemand. Vous pouvez également ajouter des traductions pour les objets de texte et créer d'autres objets dépendant de la langue.

Pour en savoir davantage, reportez-vous au *Guide d'utilisation d'IBM Cognos Report Studio*.

- Définition de la langue d'affichage d'un rapport

Vous pouvez utiliser IBM Cognos Connection pour :

- Définir des propriétés multilingues, telles qu'un nom, une infobulle et une description, pour chaque entrée du portail.
- Indiquer la langue par défaut à utiliser lors de l'exécution d'un rapport.

**Astuce :** Vous pouvez définir la langue par défaut du rapport dans la page des options d'exécution, dans les propriétés du rapport ou dans vos préférences.

- Définir une langue autre que la langue par défaut à utiliser lors de l'exécution d'un rapport.

Pour en savoir davantage, reportez-vous au *Guide d'utilisation d'IBM Cognos Connection*.

Les données s'affichent alors dans la langue et selon les paramètres régionaux définis dans

- les options du navigateur Web de l'utilisateur,
- les options d'exécution,
- les préférences d'IBM Cognos Connection.

Tout texte ajouté par les utilisateurs ou les auteurs s'affiche dans la langue utilisée lors de la saisie.

## Modélisation avec des sources de données multilingues

Pour activer l'utilisation de plusieurs langues dans un projet, vous devez configurer des sources de données pour qu'elles prennent en charge plusieurs langues.

### Sources de données relationnelles multilingues

S'agissant de sources de données relationnelles, vous pouvez permettre la prise en charge de plusieurs langues en utilisant un ou plusieurs des éléments suivants :

- Tables de base de données spécifiques de la langue

La source de données doit contenir les mêmes tables pour chaque langue prise en charge. Par exemple, si la table *Produit* prend en charge l'anglais, le français et l'allemand, la source de données comporte des tables appelées *Produit\_en*, *Produit\_fr* et *Produit\_de*.

- Colonnes spécifiques de la langue

Une table de base de données doit contenir les mêmes colonnes pour chaque langue prise en charge. Par exemple, si la table *Produit* prend en charge l'anglais, le français et l'allemand, la table comporte des colonnes appelées *NomProduit\_en*, *NomProduit\_fr* et *NomProduit\_de*.

- Lignes spécifiques de la langue

Une table de base de données doit contenir une colonne supplémentaire qui identifie la langue de chaque ligne de données, par exemple, une colonne appelée *LANG*.

Ces solutions peuvent générer des sources de données multilingues volumineuses et difficiles à gérer.

Vous pouvez modéliser un sujet de requête relationnelle unique pour représenter toutes les langues de source de données possibles en utilisant des tables de mappage des paramètres et des paramètres de session dans la définition de sujet de requête. Pour plus d'informations, voir «Création d'invites avec des macros de requête», à la page 165 et «Métadonnées multilingues», à la page 125.

**Remarque :** La syntaxe de l'expression est spécifique à la langue de conception du modèle. Si vous importez des objets à partir d'un modèle conçu dans une autre langue, vous pouvez être amené à ajuster la syntaxe de l'expression.

## Sources de données SAP BW multilingues

S'agissant de métadonnées SAP BW, vous n'avez pas besoin d'utiliser des paramètres permettant de prendre en charge la génération de rapports multilingues. En effet, SAP BW fournit automatiquement des données dans la langue qui correspond aux paramètres d'ouverture de session de l'utilisateur connecté. S'il n'existe aucune métadonnée pour la langue de l'utilisateur connecté, Framework Manager extrait les données dans la langue par défaut.

## Utilisation de macros pour modéliser des données multilingues

Vous pouvez modéliser les données multilingues stockées dans plusieurs tables, colonnes ou lignes pour chaque langue prise en charge. Vous pouvez utiliser des macros avec des tables de mappage de paramètres et des paramètres de session pour créer les dimensions ou les sujets de requête permettant d'extraire les données dans la langue préférée de la personne affichant le rapport.

L'emplacement d'un paramètre dans la définition du sujet de requête dépend de l'emplacement des données multilingues dans la source de données. Vous devez disposer d'une table de mappage des paramètres dans une macro.

Source de données	Exemple
Colonne avec une clé de langue dans une autre colonne	L'emplacement des paramètres figure dans la liste Select. <pre>Select PRODUCT_TYPE.PRODUCT_TYPE_CODE, PRODUCT_TYPE.PRODUCT_LINE_CODE, PRODUCT_TYPE.PRODUCT_TYPE_ #Language_lookup {runLocale}# as Product_type from [gossales].PRODUCT_TYPE PRODUCT_TYPE</pre>

Source de données	Exemple
Lignes dont la langue est identifiée par une colonne spéciale, telle que LANG	<p>L'emplacement des paramètres figure dans le filtre.</p> <pre> Select PRODUCT.PRODUCT_NAME, PRODUCT_MULTILINGUAL. PRODUCT_NUMBER from [gosales].PRODUCT, [gosales].PRODUCT_MULTILINGUAL Where PRODUCT.PRODUCT_NUMBER = PRODUCT_MULTILINGUAL. PRODUCT_NUMBER and (PRODUCT_MULTILINGUAL."LANGUAGE" = #sq(\$Language_lookup{\$runLocale})# </pre>

## Ajout d'une langue à un projet

Vous pouvez ajouter une langue à un projet à tout moment, par exemple, si les valeurs pour une langue n'étaient pas traduites précédemment.

Lorsque vous ajoutez une langue à un projet, IBM Cognos Framework Manager génère une nouvelle valeur pour chaque propriété multilingue de chaque objet du projet. Une propriété multilingue est une propriété de texte qui s'affiche dans un rapport, par exemple, **Nom**, **Description** et **Infobulle**.

Les nouvelles valeurs que Framework Manager affecte à ces propriétés de texte sont créées avec la valeur de la propriété d'origine précédée par le code de langue. Par exemple, si une dimension s'appelle Pays, et que vous ajoutez la langue Néerlandais, Framework Manager insère un nom dont la valeur est (nl)Pays.

Chaque projet contient deux définitions de langue.

- Langue de conception

Il s'agit de la langue dans laquelle le modèle a été créé. Cette valeur est stockée dans le modèle et ne peut pas être modifiée. Elle tient lieu de valeur de langue par défaut.

- Langue active

Il s'agit de la langue dans laquelle le contenu de modèle est actuellement affiché. Lorsque vous ouvrez un modèle, la langue active est définie sur la langue du modèle qui correspond le mieux aux paramètres régionaux et linguistique de l'ordinateur. Vous pouvez modifier cette valeur à tout moment pour votre session en cours uniquement. Dans les sessions futures, le modèle continuera de s'ouvrir dans la langue utilisée lors de la conception.

### Procédure

1. Cliquez sur **Projet, Langues, Définir les langues**.
2. Dans la zone **Langues disponibles**, sélectionnez chaque langue à ajouter et cliquez sur le bouton flèche pour la déplacer vers la zone **Langues du projet**.  
**Astuce :** Pour supprimer une langue, sélectionnez-la dans la zone **Langues du projet** et cliquez sur le bouton flèche pour la déplacer vers la zone **Langues disponibles**.

3. Si vous souhaitez modifier la langue active, dans la zone **Langues du projet**, cliquez sur une langue, puis cliquez sur **Définir en tant que valeur active, OK**. A l'invite, acceptez les modifications apportées au projet, puis cliquez sur **OK**.
4. Si vous souhaitez afficher des valeurs de propriétés multilingues dans la sous-fenêtre **Propriétés**, cliquez sur l'onglet **Langues**.

## Exportation des tables de traduction

Vous pouvez générer et exporter une table de traduction pour simplifier la tâche de traduction des objets de modèle. La table de traduction contient une liste de toutes les chaînes de texte définies pour les propriétés multilingues, telles que **Nom**, **Description**, et **Infobulle**. Les traducteurs peuvent ensuite utiliser une application externe, telle que Microsoft Excel, pour taper les informations requises dans la table.

Vous pouvez exporter une table de traduction sous la forme d'un fichier de valeurs séparées par des virgules (.csv) ou d'un fichier texte Unicode (.txt). Vous devez exporter la table de traduction en tant que fichier texte Unicode si elle contient une langue non latine ou si elle sera importée par un ordinateur dont le paramètre linguistique est différent de celui de votre ordinateur.

### Exportation d'une table de traduction

#### Procédure

1. Sélectionner les objets à exporter.
2. Cliquez sur **Projet, Langues, Exporter le fichier de traduction**.
3. Dans la zone **Langues du projet**, cliquez sur les langues à exporter, puis cliquez sur le bouton flèche pour les déplacer vers la zone **Langues à exporter**.  
Vous devez exporter la langue de conception du modèle qui utilisera la table de traduction. Par exemple, si la table de traduction doit être utilisée dans un modèle qui utilise le français comme langue de conception, vous devez exporter le français.  
IBM Cognos Framework Manager exporte le code des paramètres régionaux correspondant à chaque langue que vous sélectionnez. Si vous ne sélectionnez pas toutes les langues pour la traduction, vous devez entrer manuellement les codes de langue dans la première ligne de chaque nouvelle colonne de langue dans la table de traduction.
4. Dans la zone **Objets de modèle à exporter**, indiquez si vous souhaitez exporter tous les objets du modèle, ou uniquement les objets présélectionnés et leurs enfants.
5. Entrez l'emplacement et le nom de la table de traduction.
6. Cliquez sur **OK**.

## Importation des tables de traduction

Vous pouvez ajouter des valeurs de propriété de texte pour chaque langue définie dans votre modèle en important des données traduites à partir d'un fichier. Le fichier importé doit être une table de traduction qui a été utilisée par les traducteurs pour entrer les valeurs traduites requises.

La table de traduction doit contenir la langue de conception du modèle qui utilisera la table de traduction. La table de traduction peut contenir un sous-ensemble des langues définies pour le projet.

## Importation d'une table de traduction

### Procédure

1. Cliquez sur **Projet, Langues, Importer le fichier de traduction**.
2. Dans la zone **Langues du projet**, cliquez sur les langues de la table de traduction, puis cliquez sur les boutons flèches pour les déplacer vers les zones **Traduire à partir de** et **Traduire en**.  
Vous devez sélectionner une langue de conception pour ce modèle.
3. Dans la zone **Appliquer la traduction à**, indiquez si vous souhaitez appliquer la traduction à tous les objets du modèle, ou uniquement aux objets présélectionnés et leurs enfants.
4. Entrez l'emplacement et le nom du fichier de traduction.
5. Cliquez sur **OK**.

## Exemple - Création de projets multilingues pour des métadonnées relationnelles

Vous souhaitez créer un modèle qui peut être utilisé par des auteurs de rapport anglais, français et allemands. Vous voulez également que les studios IBM Cognos affichent automatiquement les métadonnées dans la langue requise par l'auteur du rapport.

Dans l'exemple entrepôt\_données\_VA, vous devez effectuer les opérations suivantes :

- Traduisez les métadonnées.
- Utilisez des macros pour créer un projet multilingue.

Vous pouvez modifier les dimensions et les sujets de requête pour afficher le contenu multilingue à l'aide de la table de mappage des paramètres Language\_lookup et du paramètre de session runLocale.

### Procédure

1. Ouvrez le projet entrepôt\_données\_VA.
2. Vérifiez que l'anglais, le français et l'allemand sont des langues prises en charge.
  - Cliquez sur **Projet, Langues, Définir les langues**.
  - Vérifiez que la sous-fenêtre **Langues du projet** contient Anglais, Français et Allemand.
  - Dans la sous-fenêtre **Afficheur de projet**, cliquez sur un élément de requête, puis, dans la sous-fenêtre **Propriétés**, cliquez sur l'onglet **Langues**.  
Pour le nom, la description et le texte d'infobulle, vous voyez une entrée pour chaque langue.
3. Exportez toutes les langues et tous les objets du projet dans un fichier de valeurs séparées par des virgules (.csv) nommé GOSLDW-ML.csv.
  - Cliquez sur **Projet, Langues, Exporter le fichier de traduction**.
  - Dans la zone **Langues du projet**, appuyez sur Ctrl et cliquez sur **Anglais, Français et Allemand**, puis cliquez sur la flèche vers le haut pour déplacer ces éléments vers la zone **Langues à exporter**.
  - Dans la zone **Exporter les langues dans ce fichier**, entrez l'emplacement de GOSLDW-ML.csv.
4. Ouvrez le fichier GOSLDW-ML.csv dans Microsoft Excel et traduisez les chaînes.

Notez que chaque colonne représente une langue donnée et que le fichier contient uniquement les chaînes de texte qui existent dans le modèle.

5. Dans Framework Manager, importez les fichiers traduits :
  - Cliquez sur **Projet, Langues, Importer le fichier de traduction**.
  - Dans la zone **Langues du projet**, déplacez **Français** et **Allemand** vers la zone **Traduire en**.
  - Dans la zone **Importer la table de traduction à partir de ce fichier**, entrez l'emplacement de GOSLDW-ML.csv.

6. Dans l'**Afficheur de projet**, cliquez sur la table de mappage des paramètres **Language\_lookup**.

Notez que les clés correspondent aux valeurs possibles pour le paramètre de session runLocale qui identifie la langue de l'utilisateur en cours. Ces clés sont mappées aux valeurs de langue qui sont définies dans la base de données entrepôt\_données\_VA.

7. Vérifiez que la table de mappage des paramètres contient les informations suivantes :

Clé	Valeur
de	DE
fr	FR
fr	FR

8. Sélectionnez un objet qui contient des colonnes multilingues, par exemple, la dimension Mode de commande, puis cliquez sur **Actions, Editer la définition**. L'identificateur de langue qui a été ajouté au nom d'élément de requête est remplacé par une table de mappage des paramètres et le paramètre de session runLocale :

```
Select
ORDER_METHOD_DIMENSION.ORDER_METHOD_KEY,
ORDER_METHOD_DIMENSION.ORDER_METHOD_CODE,
ORDER_METHOD_DIMENSION.ORDER_METHOD_#Language_lookup{$runLocale}
# AS ORDER_METHOD
from
[go_data_warehouse].ORDER_METHOD_DIMENSION
```

9. Pour afficher un aperçu des résultats, cliquez sur **Test, Tester l'exemple**.
10. Cliquez sur **Projet, Paramètres de session**, puis modifiez la valeur de paramètre de sessionrunLocale en fr.
11. Testez à nouveau la dimension Mode de commande pour afficher les résultats.

---

## Éléments de requête

Un élément de requête est le plus petit objet du modèle pouvant être placé dans un rapport. Il représente une instance unique d'un élément, par exemple, la date à laquelle un produit a été mis sur le marché.

Pour des métadonnées relationnelles, vous pouvez modifier les propriétés des éléments de requête :

- En définissant les propriétés **Usage** et **Agrégat ordinaire** pour refléter l'utilisation prévue de l'élément de requête
- En formatant les éléments de requête pour contrôler la manière dont les données s'affichent dans un rapport
- En identifiant une colonne comme invite et en contrôlant la manière dont les utilisateurs voient les informations d'invite

Vous pouvez également modifier les propriétés de plusieurs éléments de requête en même temps. Pour plus d'informations, voir «Remplacement de plusieurs valeurs de propriété», à la page 17.

Vous pouvez renommer un élément de requête dans la boîte de dialogue **Définition du calcul**. Le fait de renommer l'élément de requête met à jour ses références.

Comme les rapports peuvent contenir des éléments de requête différents issus d'un ou de plusieurs objets du modèle, les propriétés d'élément de requête contrôlent de nombreux aspects du rapport final. Lorsque vous créez une dimension de modèle ou un sujet de requête de modèle, les éléments de requête héritent des propriétés des éléments de requête de source de données sur lesquels ils sont basés.

Les propriétés des éléments de requête ou des mesures sont notamment les suivantes :

Propriété d'élément de requête	Description
Nom	Nom de l'élément de requête ou de la mesure.
Description	Description de l'élément de requête ou de la mesure.
Dernière modification	Date à laquelle l'élément de requête ou la mesure a été modifié pour la dernière fois. La propriété est automatiquement mise à jour avec l'heure et la date en cours.
Dernière modification effectuée par	Utilisateur qui a modifié en dernier l'élément de requête ou la mesure. Cette propriété est automatiquement mise à jour lorsque l'élément est modifié.  Sa valeur est le nom d'utilisateur de connexion en cours.
Commentaires de modèle	Permet d'ajouter des commentaires internes concernant le modèle. Ces informations sont utilisées dans la boîte de dialogue <b>Analyser l'impact de publication</b> et dans le <b>rapport de modèle</b> . Les commentaires ne sont pas accessibles aux utilisateurs du pack.
Infobulle	Description pouvant s'afficher dans le pack publié pour les utilisateurs.

Propriété d'élément de requête	Description
Expression	<p>Permet de créer des calculs intégrés qui fournissent aux utilisateurs les valeurs calculées qu'ils utilisent régulièrement.</p> <p>Cette propriété est destinée aux mesures uniquement.</p> <p><b>Remarque :</b> La propriété <b>Expression</b> n'est pas utilisée par SAP BW.</p>
Nom externe	Nom qui s'affiche dans la source de données.
Est masqué	<p>Indique s'il faut masquer ou afficher l'élément de requête ou la mesure dans le pack publié.</p> <p>Même si <b>Est masqué</b> est défini sur <b>True</b> et que l'élément de requête ou la mesure est invisible pour les utilisateurs, l'élément de requête ou la mesure figure quand-même dans le pack publié car ils peuvent être requis par d'autres objets du modèle. Vous ne voyez pas l'élément de requête ou la mesure dans l'assistant de <b>publication de pack</b>.</p> <p>Par exemple, un calcul peut utiliser un élément de requête masqué.</p>
Usage	<p>Utilisation prévue pour les données représentées par l'élément de requête.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p>
Format	Manière dont les informations apparaissent dans un rapport.
Devise	<p>Devise utilisée.</p> <p>Cette propriété ne peut pas être modifiée dans la sous-fenêtre <b>Propriétés</b>. Utilisez la propriété <b>Format</b> pour modifier la devise.</p>
Type de données	<p>Type de données défini dans la source de données.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Précision	<p>Nombre total de chiffres.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>



Propriété d'élément de requête	Description
Echelle	<p>Nombre de chiffres représentés dans l'échelle.</p> <p>Par exemple, vous pouvez afficher des nombres en milliers de sorte que 100 000 corresponde à 100 000 000.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Taille	<p>Taille de l'élément de requête ou de la mesure.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
NULL admis	<p>Indique si l'élément de requête ou la mesure peut contenir une valeur NULL.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Type d'affichage	<p>Indique comment l'élément de requête est affiché.</p> <p>La valeur de colonne peut s'afficher dans les studios IBM Cognos sous la forme d'une image, d'un lien ou d'une valeur.</p> <p>La valeur par défaut est value.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p>
Type MIME	<p>Format utilisé par la valeur de colonne.</p> <p>Par exemple, si <b>Type d'affichage</b> est définie sur picture, <b>Type MIME</b> peut être jpeg.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p> <p><b>Remarque :</b> La propriété <b>Type MIME</b> propriété n'est pas utilisée par SAP BW.</p>
Informations d'invite	Comportement de l'invite.
Agrégat ordinaire	Type d'agrégation associé à l'élément de requête, la mesure ou le calcul dans le pack publié.

Propriété d'élément de requête	Description
Règles d'agrégation	<p>Pour les métadonnées relationnelles modélisées de façon dimensionnelle, les règles pour l'agrégation semi-additive.</p> <p>Pour les métadonnées SAP BW, la propriété <b>Règles d'agrégation</b> est accessible en lecture seule.</p>
Règle d'allocation	<p>Indique le type d'allocation défini pour la mesure.</p> <p>La valeur <b>par défaut</b> indique que l'allocation constante est utilisée dans les requêtes de type Liste et l'allocation unique est utilisée dans les requêtes de tableau croisé. La valeur <b>constante</b> indique que l'allocation constante est utilisée dans toutes les requêtes.</p>
N'est pas triable	<p>Indique si les valeurs de cet élément de requête peuvent être triées.</p> <p>Cette propriété est destinée aux éléments de requête contenant des objets volumineux comme les objets BLOB.</p>

## Modification du mode d'agrégation des éléments de requête

Vous pouvez modifier la façon dont certains éléments de requête et certaines mesures sont agrégés dans les rapports. IBM Cognos Framework Manager applique des règles d'agrégation lorsque les utilisateurs créent un rapport qui récapitule un élément de requête ou une mesure.

Lorsque vous importez des métadonnées, Framework Manager attribue des valeurs aux propriétés **Usage** et **Agrégat ordinaire** pour les éléments de requête et les mesures en fonction de leur type d'objet. La propriété **Usage** identifie l'utilisation prévue des données représentées par l'élément de requête («Propriété Usage», à la page 140). La propriété **Agrégat ordinaire** identifie le type d'agrégation appliqué à l'élément de requête ou la mesure («Propriété Agrégat ordinaire», à la page 141). Les utilisateurs peuvent redéfinir les valeurs de la propriété **Agrégat ordinaire**. Pour les mesures semi-additives, vous pouvez spécifier des règles d'agrégation supplémentaires en modifiant la propriété **Règles d'agrégation** («Définition de règles d'agrégation pour des mesures semi-additives», à la page 141).

Lorsque vous modifiez la propriété **Agrégat ordinaire**, vous pouvez sélectionner des valeurs qui ne sont pas disponibles via l'importation, par exemple, Moyenne et Maximum. Vous devez comprendre ce que représentent les données pour savoir quelle valeur d'agrégat est nécessaire. Par exemple, si vous agrégez un numéro de composant, les seules valeurs d'agrégat qui s'appliquent sont Nombre, Nombre (éléments distincts), Maximum et Minimum.

En règle générale, préférez l'utilisation de la propriété **Agrégat ordinaire** à l'ajout de fonctions récapitulatives dans les expressions. Utilisez des fonctions

récapitulatives dans les expressions si le récapitulatif requis n'est pas pris en charge par une propriété d'agrégat, ou si l'expression est trop complexe pour être gérée par les propriétés des éléments.

La méthode suivante de comptage des valeurs distinctes, par exemple des clients, garantit la cohérence des résultats dans les listes, les tableaux croisés et les graphiques. Il est recommandé de compter des éléments distincts sur une clé métier qui identifie les valeurs de façon unique plutôt que sur une valeur de chaîne. Cette opération prévient la perte involontaire des lignes uniques contenant un nom identique.

L'exemple suivant compte les clients distincts dans un modèle relationnel. Créez une copie de **Clé client** dans le sujet de requête **Client**, nommez-la **Nombre de clients**, définissez la propriété **Utilisation** sur **Fait** et la propriété **Agrégat ordinaire** sur **Nombre (éléments distincts)**. Il est déconseillé de créer un élément de requête avec l'expression `Count (distinct [Customer Key])`, car le comportement d'un élément de requête peut être imprévisible selon les scénarios. Si un formatage est requis, définissez le **type de format** sur **Nombre** et définissez les propriétés de **format** adéquates, par exemple le nombre de décimales ou le séparateur de milliers.

Dans un modèle DMR (Dimensionally Modeled Relational), ajoutez l'élément de requête **Nombre de clients** créé dans le sujet de requête **Client** à une dimension de mesure. La propriété **Agrégat ordinaire** et les paramètres de la propriété **Format** sont hérités. Une autre solution consiste à ajouter la **Clé client** d'origine du sujet de requête **Client** à la dimension de mesure, à définir la propriété **Agrégat ordinaire** sur **Nombre (éléments distincts)**, et à définir le format nécessaire.

## Règles de définition de propriétés pour des dimensions

IBM Cognos Framework Manager utilise les règles suivantes pour définir les propriétés **Usage** et **Agrégat ordinaire**.

Objet	Propriété Usage	Propriété Agrégat ordinaire
Élément de requête dans une dimension ordinaire	Attribut	Non pris en charge
Élément de requête dans une dimension de mesure	Identificateur	Nombre
Mesure dans une dimension de mesure	Fait	Automatique si la mesure est un calcul Somme si la mesure n'est pas un calcul

Si la mesure est semi-additive, utilisez la propriété **Règles d'agrégation** pour définir les règles d'agrégation semi-additive. Voir «Définition de règles d'agrégation pour des mesures semi-additives», à la page 141.

## Règles de définition de propriétés pour des sujets de requête

Vous pouvez modifier les propriétés **Usage** et **Agrégat ordinaire** pour tous les types de sujet de requête. Les paramètres de ces propriétés sont basés sur des caractéristiques telles que le type de données et la participation à des clés et des relations.

Pour les sujets de requête de modèle, IBM Cognos Framework Manager utilise les paramètres des sujets de requête sous-jacents. Si le sujet de requête de source n'utilise pas ces propriétés, les règles pour les sujets de requête de source de données et de procédure stockée sont appliquées.

Pour les sujets de requête de source de données ou de procédure stockée, Framework Manager utilise les règles suivantes pour définir les propriétés **Usage** et **Agrégat ordinaire** lors de l'importation des sujets de requête.

Objet	Propriété Usage	Propriété Agrégat ordinaire
L'élément de requête fait partie d'une clé dans un déterminant	Identificateur	Nombre
L'élément de requête participe à une relation	Identificateur	Nombre
L'élément de requête a le type de données date ou heure	Identificateur	Nombre
L'élément de requête a le type de données numérique ou intervalle de temps	Fait	Automatique si l'élément est un calcul Somme si l'élément n'est pas un calcul
Aucune des règles ci-dessus ne s'applique	Attribut	Non pris en charge

## Règles de définition de propriétés pour des calculs

IBM Cognos Framework Manager utilise un certain nombre de règles pour définir les propriétés **Usage** et **Agrégat ordinaire** pour les calculs.

### Règles d'interprétation des agrégations calculées :

Le paramètre Calculé de la propriété Agrégat ordinaire commande l'ordre des opérations dans les calculs.

Dans Framework Manager, le type d'agrégat Calculé ne s'applique qu'aux :

- Calculs autonomes
- Calculs basés sur les mesures d'une dimension et qui y sont intégrés

Il n'est pas pris en charge par les calculs imbriqués dans des sujets de requête.

Pour plus d'informations, voir «Définition de l'ordre des opérations pour les calculs de modèles», à la page 344.

Framework Manager utilise les règles suivantes pour interpréter le type d'agrégat Calculé dans la propriété **Agrégat ordinaire**.

- Les fonctions d'agrégation standard (moyenne, nombre, maximum, minimum, écart-type, somme, variance) et les références à des sujets de requête de modèle sont agrégées en premier. Les opérations restantes sont ensuite appliquées au résultat de l'agrégation.

Par exemple, pour diviser le débit par le crédit pour chaque ligne, le code SQL ressemble à ceci :

```
Select
customer, debt, credit, debt/credit as
percent_debt from x
```

Pour effectuer une agrégation pour tous les clients, le code SQL se présente comme suit:

```
Select sum(debt), sum(credit), sum(debt)/sum(credit)
as percent_debt from (Select customer, debt, credit from x)
```

- Si l'élément de requête dans le calcul est un fait et que le type d'agrégation pour l'élément de requête est moyenne, nombre, maximum, minimum, ou somme, le type d'agrégation de l'élément de requête est utilisé.
- Si l'élément de requête dans le calcul n'a aucun type d'agrégation défini, le type d'agrégation minimum est appliqué dans la requête. Le code SQL ne peut pas avoir un paramètre agrégation défini sur Aucun.

- Les fonctions d'agrégation sont interprétées comme si elles étaient appliquées à une valeur dans une ligne unique lorsque ces fonctions sont utilisées dans le contexte détaillé. Par exemple, la propriété Groupement et récapitulatif automatiques d'un rapport Report Studio est définie sur false.

- L'agrégation d'un élément de requête est basée sur l'expression d'agrégation dérivée de la définition de l'élément.

Par exemple, vous souhaitez effectuer le total de ce calcul autonome :

```
[namespace].[Company].[debt]
/ [namespace].[Company].[credit]
```

Le calcul est agrégé avec cette expression :

```
Total([namespace].[Company].[debt]) / Total([namespace].[Company].[credit])
```

- Les agrégats scalaires, également appelés agrégats cumulatifs, de classement et mobiles, sont calculés pour la granularité du rapport, sauf si la clause For est explicitement indiquée.

- La granularité des fonctions d'agrégation est définie par groupe pour les déterminants ou par clés de niveaux correspondants dans le cube.

Par exemple, Rank([namespace].[Company].[debt]) est interprété comme Rank([namespace].[Company].[debt] for Report).

### Règles permettant de déterminer le type d'agrégation automatique :

Pour les calculs autonomes et intégrés, IBM Cognos Framework Manager utilise une seule de ces règles pour déterminer le type d'agrégation.

Calcul	Type d'agrégation
Basé sur une expression contenant une fonction d'agrégation telles que moyenne, maximum, minimum ou somme	Calculé

Calcul	Type d'agrégation
Comporte une opération if-then-else et la condition if fait référence à des éléments de faits	Calculé
Fait référence à un calcul utilisant un type autre que Non pris en charge	Calculé
A un type d'agrégation autre que Non pris en charge	Calculé
Basé sur une expression qui fait référence à un sujet de requête de modèle dont l'utilisation est définie sur Fait et dont le type d'agrégation est moyenne, maximum, minimum ou somme, mais l'élément de requête n'utilise pas de fonction d'agrégation	Récapituler Voir ci-dessous les règles pour Récapituler.
Aucune de ces règles ne s'applique	Non pris en charge

#### Règles permettant de déterminer le type d'agrégation récapitulative :

Pour les calculs autonomes et intégrés, IBM Cognos Framework Manager utilise une seule de ces règles pour déterminer le type d'agrégation.

Calcul	Type d'agrégation
Fait de contenant uniquement une référence à un élément de requête dont le type d'agrégation est moyenne, nombre, maximum, minimum ou somme	Utilise le type d'agrégation de l'élément de requête
Numérique ou un type d'intervalle	Somme
Heure, date-heure ou type de date	Maximum
Aucune de ces règles ne s'applique	Nombre

#### Propriété Usage

La propriété **Usage** identifie l'utilisation prévue des données représentées par chaque élément de requête. Lors de l'importation, la propriété **Usage** est définie selon le type de données que les éléments de requête représentent dans la source de données.

Vous devez vérifier si cette propriété est définie correctement. Par exemple, si vous importez une colonne numérique qui participe à une relation, la propriété **Usage** est définie sur **identificateur**. Vous pouvez modifier la propriété.

Pour les éléments de requête relationnels, la valeur de la propriété **Usage** dépend du type d'objet de base de données sur lequel est basée la requête.

Propriété Usage	Objet de base de données	Description
Identificateur	key, index, date, datetime	Représente une colonne utilisée pour grouper ou récapituler les données dans une colonne Fait avec laquelle elle a une relation.  Représente également une colonne indexée.  Représente également une colonne de type date ou heure.
Fait	numeric, timeinterval	Représente une colonne qui contient des données numériques qui peuvent être groupées ou récapitulées, par exemple, Coût du produit.
Attribut	string	Représente une colonne qui n'est ni Identificateur, ni Fait, par exemple, Description.

### Propriété Agrégat ordinaire

La propriété **Agrégat ordinaire** identifie le type d'agrégation pour l'élément de requête ou le calcul lorsque vous le publiez. Les utilisateurs peuvent utiliser ce paramètre par défaut pour effectuer des calculs sur des groupes de données, ou appliquer un autre type d'agrégation.

Par exemple, si la valeur de la propriété **Agrégat ordinaire** de l'élément de requête Quantité est Somme et que le groupement se fait par nom de produit dans le rapport, la colonne Quantité du rapport montre la quantité totale de chaque produit.

Les types d'agrégation suivants sont pris en charge pour les sources de données relationnelles :

- Automatique
- Moyenne
- Calculé
- Nombre
- Nombre (éléments distincts)
- Maximum
- Minimum
- Somme

### Définition de règles d'agrégation pour des mesures semi-additives

Pour les mesures de métadonnées relationnelles modélisées de façon dimensionnelle, vous pouvez définir une règle d'agrégation pour chaque dimension associée. Ces règles viennent s'ajouter à la propriété **Agrégat ordinaire** et sont utilisées pour indiquer comment les mesures semi-additives doivent être agrégées en fonction des informations issues de cette dimension. Une mesure semi-additive est une mesure qui doit être additionnée pour certaines dimensions, mais pas toutes. Pour les dimensions sur lesquelles la mesure n'est pas additive, une règle d'agrégation différente doit être spécifiée.

Si une expression de mesure contient une fonction d'agrégation, la règle d'agrégation pour cette mesure est ignorée.

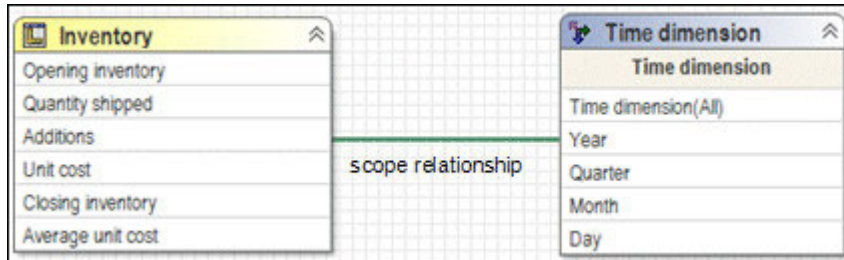
Les règles d'agrégation sont appliquées dans l'ordre suivant :

- La propriété **Agrégat ordinaire** est appliquée aux dimensions qui sont incluses dans la requête mais qui ne sont pas référencées dans la boîte de dialogue **Règles d'agrégation**
- L'agrégation définie dans la boîte de dialogue **Règles d'agrégation** est appliquée aux dimensions spécifiées dans l'ordre dans lequel vous avez indiqué les règles
- L'agrégation de niveau rapport indiquée dans la requête est appliquée

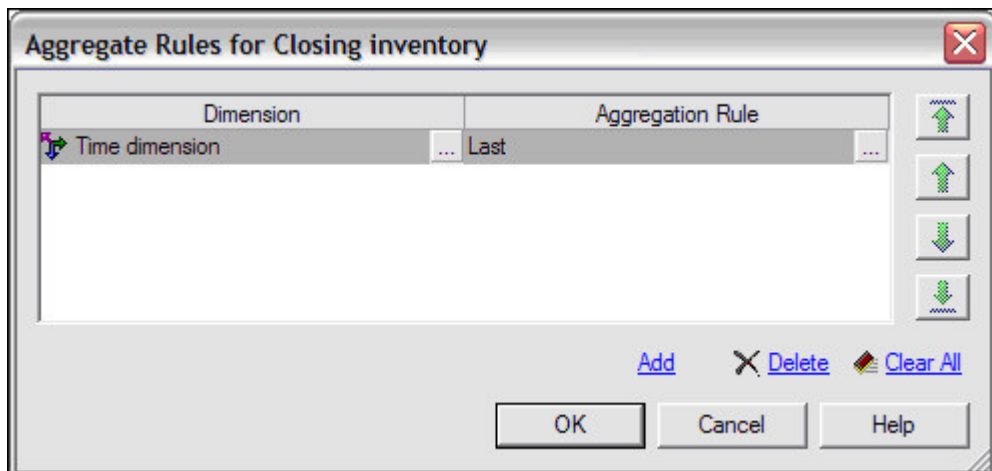
Par exemple, le stock est enregistré dans l'entrepôt de données avec des soldes d'ouverture et de clôture à une granularité mensuelle. Si vous avez besoin de générer des rapports sur le stock pour d'autres périodes, vous appliquez une agrégation qui prend la valeur d'intervalle appropriée au sein de la période. Pour un solde d'ouverture, la valeur est le dernier solde du mois précédent. Pour un solde de clôture, la valeur est le dernier solde du mois en cours. Dans cet exemple, la mesure de stock comprend une propriété **Agrégat ordinaire** de total et une règle d'agrégation pour la dimension de temps avec la valeur Dernier.

Vous ne pouvez avoir qu'une règle d'agrégation par dimension.

**Remarque :** Il doit exister une relation directe entre les sujets de requêtes sous-jacents à partir desquels la mesure et la dimension sont créées. Par exemple, examinons la dimension de mesure Inventory (Stock) et la dimension Time (Temps) :

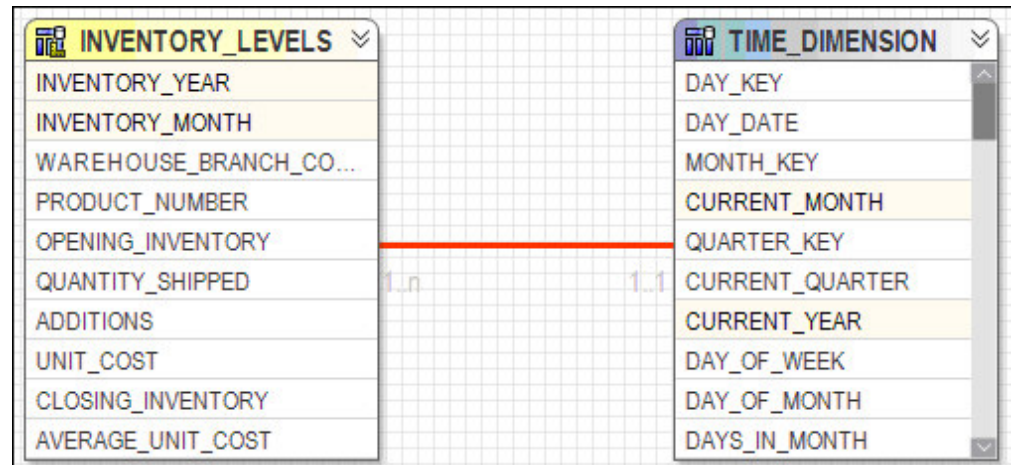


Vous souhaitez définir une règle d'agrégation pour le stock de clôture qui génère le dernier stock de clôture enregistré pour une période donnée :





Pour qu'IBM Cognos applique la règle d'agrégation Dernier (Last) sur le stock de clôture (Closing inventory), basée sur la dimension de temps (Time), il doit exister une relation directe entre les sujets de requête sous-jacents utilisés pour créer la dimension de temps et la mesure de stock de clôture. Cela vient s'ajouter à la relation de portée définie entre la dimension de mesure Inventory et la dimension Time :



S'il n'existe pas de relation, l'exécution d'une requête qui contient la mesure Closing inventory génère l'erreur suivante :

RQP-DEF-0564 Impossible de déterminer le joint de fait de la dimension semi-additive '[Dimensional view].[Time dimension]' pour la mesure '[Dimensional view].[Inventory].[Closing inventory]'. Vérifiez qu'une relation est définie entre les sujets de requêtes sous-jacents à partir desquels les dimensions sont créées.

Les mesures semi-additives ne sont pas prises en charge pour les données fragmentées. Les données fragmentées sont des données multidimensionnelles dans lesquelles la plupart des combinaisons de valeurs de dimension ne sont pas associées à des données de fait.

Pour les métadonnées SAP BW, la propriété **Règles d'agrégation** est accessible en lecture seule. La propriété **Semi-agrégat** est utilisée à la place.

### Procédure

1. Cliquez sur la mesure à utiliser.
2. Dans la sous-fenêtre **Propriétés**, cliquez sur les points de suspension (...) dans la propriété **Règles d'agrégation**.
3. Pour ajouter une dimension pour cette mesure, cliquez sur **Ajouter** et sélectionnez la dimension.  
Vous pouvez inclure un raccourci vers une dimension s'il existe une relation de portée entre la dimension de mesure et la dimension ordinaire.
4. Pour indiquer l'opération d'agrégation, cliquez sur les points de suspension (...) dans la zone **Règles d'agrégation**.

Les opérations suivants sont prises en charge :

- Somme
- Minimum
- Maximum

- Moyenne
  - Premier
  - Dernier
5. Si vous souhaitez modifier l'ordre des règles d'agrégation, utilisez les flèches.
  6. Cliquez sur **OK**.

## Résultats

Une fois que vous avez défini une règle d'agrégation, l'icône correspondant à la mesure est modifiée.

### Propriété Semi-agrégat

Pour les métadonnées relationnelles, la valeur de la propriété **Semi-agrégat** est définie sur non pris en charge et est en lecture seule.

Si la valeur est définie sur **non pris en charge** dans IBM Cognos Framework Manager, le comportement de semi-agrégation est ignoré dans les studios IBM Cognos.

La propriété **Semi-agrégat** ne sera pas prise en charge dans les futures éditions. Utilisez plutôt la propriété **Règles d'agrégation** pour les mesures semi-additives.

### Détermination des règles d'utilisation et d'agrégation

Lorsque vous utilisez les commandes **Déterminer l'utilisation** et **Déterminer les règles d'agrégation** dans IBM Cognos Framework Manager, vous réinitialisez les propriétés **Usage** et **Agrégat ordinaire** à leurs valeurs par défaut. Cela écrase toutes les modifications que vous avez apportées à ces propriétés.

Lors de la génération des valeurs d'agrégation, Framework Manager affecte une valeur qui est basée sur la valeur de la propriété **Usage** et le type d'objet correspondant.

Valeur de la propriété Usage	Valeur de la propriété Agrégat ordinaire
Identificateur	Nombre
Attribut	Non pris en charge
Fait	Somme

### Procédure de détermination de l'utilisation et des règles d'agrégation :

#### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, cliquez sur un ou plusieurs sujets de requête ou dimensions.
2. Dans la sous-fenêtre **Propriétés**, cliquez sur l'onglet **Propriétés**.
3. Modifiez la propriété **Utilisation** en **inconnu**.
4. Modifiez la propriété **Agrégat ordinaire** en **non pris en charge**.
5. Cliquez sur **Outils, Déterminer l'utilisation**.
6. Cliquez sur **Outils, Déterminer les règles d'agrégation**.

## Formatage des éléments de requête

Vous pouvez spécifier la manière dont les valeurs d'élément de requête s'affichent dans les rapports. Utilisez la propriété **Format** pour choisir un type de format, tel qu'un texte, une date ou une devise. Chaque type de format contient des propriétés qui définissent de manière plus précise le mode d'affichage des données.

Par exemple, vous pouvez affecter le type de format **Devise** à un élément de requête numérique, puis utiliser la propriété **Nombre de décimales** de la boîte de dialogue **Format de données** pour indiquer le nombre de positions décimales qui s'affichent dans les rapports.

Certains caractères sont sensibles à la langue et s'affichent correctement uniquement lorsque vos paramètres régionaux prennent en charge la police applicable. Par exemple, pour que le symbole de la devise japonaise s'affiche correctement, la valeur Japonais doit être affectée à vos paramètres régionaux.

Si IBM Cognos Framework Manager n'affiche pas la devise que vous demandez, prenez soin d'installer les modules de langue appropriés pour permettre la prise en charge des symboles de devise. Par exemple, pour que le symbole de la devise indienne (roupie) puisse s'afficher, vous devez exécuter un système d'exploitation ou installer un module de langue qui peuvent afficher ce symbole. Le système d'exploitation japonais ou la langue japonaise permettent d'afficher le symbole de la devise indienne.

Vous pouvez définir des propriétés pour plusieurs éléments de requête en même temps. Toutefois, si les éléments de requête possèdent différents types de format, toutes les propriétés déjà spécifiées sont remplacées et les valeurs par défaut de la source de données sont utilisées. Si les types de format d'origine des éléments de requête sélectionnés sont identiques, toutes les propriétés des éléments de requête sélectionnés sont définies de manière identique.

Par exemple, pour utiliser le même séparateur décimal pour deux éléments de requête et maintenir un nombre de décimales différent, chaque élément de requête doit être modifié individuellement. Si deux éléments de requête sont modifiés en même temps, toutes les propriétés, y compris le nombre de positions décimales, sont définies de manière identique pour ces deux éléments de requête.

### Procédure de formatage d'un élément de requête

#### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, cliquez sur l'élément de requête à modifier.
2. Dans l'onglet **Propriétés** de la sous-fenêtre **Propriétés**, cliquez sur la propriété **Format**.
3. Sélectionnez le **type de format** approprié.
4. Dans la zone **Propriétés**, sélectionnez ou entrez la valeur de propriété appropriée.
5. Cliquez sur **OK**.

## Définition des contrôles d'invite

Les invites permettent à vos utilisateurs de retrouver facilement les informations dont ils ont besoin dans un rapport. Les invites sont généralement définies dans des rapports. Vous pouvez toutefois modifier leur comportement dans les studios en changeant la définition des dimensions ou des sujets de requête dans le modèle.

Ceci est utile pour les éléments de requête, tels que ProductTypeCode, dont les valeurs ne sont pas affichées dans un rapport, mais sont utiles pour le filtrage des données. En règle générale, il est préférable de définir des invites de saisie au clavier dans les rapports afin de pouvoir utiliser des fonctions d'invite supplémentaires. Cependant, vos utilisateurs ne peuvent pas modifier certaines variables. Pour ces variables, vous pouvez utiliser IBM Cognos Framework Manager au lieu des rapports pour définir des invites de saisie au clavier.

Les propriétés Informations d'invite définies dans Framework Manager vous offrent la possibilité de contrôler le filtrage et les invites par défaut. Les propriétés sont utilisées par :

- Query Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- L'outil Créer une page d'invite de Report Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- Des invites générées dans Report Studio, pour définir l'utilisation et l'affichage d'éléments dans l'invite et la requête d'invite.

La syntaxe permettant d'utiliser une invite en tant que valeur est la suivante :  
?<PromptName>?

Vous pouvez utiliser des invites dans

- des tables de mappage des paramètres ;
- des paramètres de session ;
- des arguments de procédure stockée ;
- des expressions, y compris des filtres, des calculs et des relations.

Par exemple, une procédure stockée renvoie toutes les lignes avec un numéro de produit spécifique. Au lieu d'utiliser le numéro de produit comme argument pour la procédure stockée, vous pouvez utiliser une invite, telle que ?Product\_Number?.

Pour obtenir des exemples d'invites, voir «Création d'invites avec des macros de requête», à la page 165 et «Exemple - Utilisation d'invites avec une procédure stockée», à la page 84.

Pour plus d'informations sur la création d'invites, voir le document Report Studio *User Guide*.

## Procédure de définition d'un contrôle d'invite

### Procédure

1. Cliquez sur l'élément de requête.
2. Dans la sous-fenêtre **Propriétés**, cliquez sur l'onglet **Propriétés**.
3. Cliquez sur le signe plus (+) en regard de la propriété **Informations d'invite**.  
Il s'agit d'une propriété d'élément de requête composée.
4. Modifiez les propriétés suivantes selon le comportement dont vous avez besoin.

Objectif	Propriété
Définir le type de contrôle d'invite qui est généré lorsque le rapport est exécuté.	Type d'invite

Objectif	Propriété
Définir l'invite générée dans le cadre d'une série d'invites en cascade générées.	Cascade sur la référence d'élément
Indiquer l'élément de requête qui s'affiche pour l'utilisateur du rapport dans l'invite.  Les valeurs de l'invite sont les valeurs de données de l'élément de requête.  Chaque valeur de l'invite est associée à une valeur de l'élément de requête spécifié dans la propriété <b>Utiliser la référence d'élément</b> .	Afficher la référence d'élément
Indiquer l'élément de requête qui est transmis de l'invite au filtre.  Chaque valeur est associée à une valeur de l'élément de requête spécifié dans la propriété <b>Afficher la référence d'élément</b> .	Utiliser la référence d'élément
Indiquer l'élément de requête utilisé dans l'expression de filtre pour extraire des données.	Filtrer la référence d'élément

### Propriété de type d'invite

La propriété **Type d'invite** définit le type de contrôle d'invite généré lorsque le rapport est exécuté, par exemple, une zone d'édition ou une liste déroulante.

La valeur par défaut de cette propriété est **Déterminé par le serveur**.

**Remarque:** Les types d'invite définis sur des attributs sont désormais traités. L'utilisateur du rapport verra l'invite qui correspond au type d'invite de l'attribut. Comme les types d'invite sur des attributs n'étaient pas traités dans la version précédente, certaines différences peuvent avoir lieu.

Valeur	Contrôle d'invite
Déterminé par le serveur	Le type de contrôle d'invite dépend d'informations sur le serveur, telles que le type de données de l'élément de requête.
Zone d'édition	Zone de texte simple. Si le type de données de la colonne est date ou date-heure, cette valeur génère un contrôle de date ou de date-heure, ainsi que la zone de texte.
Sélectionner une date	Contrôle de date avec une interface de calendrier.
Sélectionner une date/heure	Contrôle de date-heure avec une interface de calendrier.  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.

Valeur	Contrôle d'invite
Sélectionner un intervalle	Contrôle d'intervalle de date-heure  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.
Sélectionner une heure	Contrôle d'heure qui filtre les données en fonction de la période sélectionnée. Par exemple, si vous définissez une invite Sélectionner une heure pour Heure de la commande, l'utilisateur peut utiliser le contrôle d'heure pour afficher toutes les commandes passées après 1:00, ou toutes les commandes passées entre 10:00 et 11:00.  Si vous faites référence à un membre d'heure, vous devez utiliser les valeurs exactes uniquement.  Si vous utilisez une plage, les limites de la plage doivent correspondre à des valeurs de la source de données.
Sélectionner une valeur	Liste déroulante.
Sélectionner avec recherche	Contrôle de liste permettant à des utilisateurs de rechercher des valeurs.  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.
Sélectionner avec arborescence	Contrôle d'invite hiérarchique pour les invites qui sont basées sur un noeud hiérarchique.

### Propriété Cascade sur la référence d'élément

La propriété **Cascade sur la référence d'élément** indique que l'invite générée fait partie d'une série d'invites en cascade générées. L'élément de requête que vous référencez dans cette propriété est l'élément parent de la cascade. Le système invite l'utilisateur à entrer l'élément en cascade avant l'élément de requête en cours.

Par exemple, si vous souhaitez inviter l'utilisateur à saisir la ligne de produits, puis le produit au sein de la ligne sélectionnée, définissez la propriété **Cascade sur la référence d'élément** de l'élément de requête Produit sur Ligne de produits.

### Propriétés Afficher la référence d'élément et Utiliser la référence d'élément

La propriété **Afficher la référence d'élément** indique quel élément de requête est affiché pour l'utilisateur dans l'invite. La propriété **Utiliser la référence d'élément** indique quel élément de requête est transmis de l'invite au filtre. Chaque valeur de la liste des éléments d'affichage est associée à une valeur de l'élément de requête spécifié dans la propriété **Utiliser la référence d'élément**.

Par exemple, vous souhaitez que l'invite Nom de pays s'affiche lors de l'utilisation de Code de pays pour extraire des données. Affectez la valeur Nom de pays à la propriété **Afficher la référence d'élément** et la valeur Code de pays à la propriété **Utiliser la référence d'élément**. L'invite pour Nom de pays permet à l'utilisateur du rapport de sélectionner plus facilement les valeurs requises. Cependant, l'utilisation de Code de pays dans le filtre s'avère plus efficace pour l'extraction de données.

Ces propriétés sont utilisées par :

- Query Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- L'outil Créer une page d'invite de Report Studio, pour définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- Des invites générées dans Report Studio, pour définir l'utilisation et l'affichage d'éléments dans l'invite et la requête d'invite.

**Remarque :** les valeurs des propriétés **Utiliser la référence d'élément** et **Filtrer la référence d'élément** doivent être compatibles. Sinon, les résultats de l'utilisateur du rapport peuvent être inattendus. Pour plus d'informations, voir «Propriété Filtrer la référence d'élément».

Valeur par défaut : si aucune valeur n'est définie, les propriétés prennent par défaut le nom de l'élément de requête.

Ces propriétés sont utilisées uniquement pour les contrôles d'invite pilotés par des données dont la propriété **Type d'invite** a pour valeur **Sélectionner une valeur** ou **Sélectionner avec recherche**.

### **Propriété Filtrer la référence d'élément**

La propriété **Filtrer la référence d'élément** identifie l'élément de requête utilisé lorsque Report Studio ou Query Studio génère un filtre. Cette propriété facilite la création de requêtes plus efficaces en faisant en sorte qu'un filtre utilise une colonne numérique indexée plutôt qu'une colonne de chaîne non indexée.

Par exemple, un auteur de rapport souhaite créer un filtre pour l'élément de requête Nom de pays. Vous définissez la propriété **Filtrer la référence d'élément** de sorte que l'élément Code de pays soit utilisé au lieu de l'élément Nom de pays dans n'importe quel filtre qui utilise l'élément de requête Nom de pays.

Dans un autre exemple, un auteur de rapport veut créer un filtre pour l'élément de requête Code de pays qui s'affiche dans la table Commandes. Vous souhaitez que ce filtre utilise l'élément Code de pays de la table Pays car celle-ci contient moins de lignes à lire. Par conséquent, vous affectez Country.Country Code à la propriété **Filtrer la référence d'élément** dans le modèle.

Cette propriété est utilisée par :

- Query Studio, pour créer une expression de filtre.
- L'outil Créer une page d'invite de Report Studio, pour créer une expression de filtre.

Valeur par défaut : si aucune valeur n'est définie, la propriété prend par défaut le nom de l'élément de requête.

### **Utilisation des propriétés Filtrer la référence d'élément et Utiliser la référence d'élément**

Les valeurs des propriétés **Filtrer la référence d'élément** et **Utiliser la référence d'élément** doivent être compatibles. La valeur de la propriété **Utiliser la référence d'élément** doit correspondre à un type attendu par la propriété **Filtrer la référence d'élément**. Sinon, les résultats de l'utilisateur du rapport peuvent être inattendus. Cela peut se produire lorsqu'un utilisateur de rapport crée un filtre sans créer de page d'invite.

Dans un exemple de modèle, la propriété **Utiliser la référence d'élément** prend la valeur Numéro d'employé et la propriété **Filtrer la référence d'élément** prend la valeur Nom d'employé. Dans Report Studio, un auteur de rapport crée le filtre suivant sans créer de page d'invite :

```
Reference.EmployeeName in ?parm1?
```

Report Studio génère automatiquement des invites lorsque vous créez un filtre sans créer de page d'invite. Etant donné que l'invite est générée, Report Studio utilise les propriétés Informations d'invite de l'élément de requête Nom d'employé dans le modèle Framework Manager.

La propriété **Utiliser la référence d'élément** indique que les valeurs transmises au filtre sont des numéros d'employé. La propriété **Filtrer la référence d'élément** filtre les données en fonction de la valeur Nom d'employé. Le filtre se présente comme suit : `Reference].[Employee Name] in ("1", "2")`. Dans la mesure où il n'existe aucune valeur "1" ou "2" pour le nom d'employé, le rapport est vide.

### **Utilisation de la propriété Filtrer la référence d'élément pour les métadonnées relationnelles modélisées de façon dimensionnelle**

Pour les métadonnées relationnelles modélisées de façon dimensionnelle, la propriété **Informations d'invite** est spécifiée sur l'attribut avec le rôle `_memberCaption`, et non pas sur le niveau. Bien qu'elles soient définies sur l'attribut, les propriétés **Informations d'invite** sont traitées comme si elles étaient spécifiées sur le niveau. Par défaut, lorsque le niveau est inclus dans un rapport, les utilisateurs sont invités à entrer des noms uniques de membre à l'invite de niveau. Pour entrer des valeurs de légende à la place, définissez la propriété **Filtrer la référence d'élément** de l'attribut sur lui-même. Lorsque le filtre obtenu par invite est appliqué, les valeurs filtrées sont basées sur les valeurs d'attributs.

Par exemple, le niveau Ligne de produits possède un attribut Ligne de produits doté d'un rôle `_memberCaption`. Si la valeur de propriété **Filtrer la référence d'élément** a pour valeur Ligne de produits, les utilisateurs de rapport sont invités à entrer des valeurs Ligne de produits. Si la valeur de propriété **Filtrer la référence d'élément** est vide, les utilisateurs sont invités à entrer des noms uniques de membre.

**Remarque** : vous ne devez pas utiliser la propriété **Filtrer la référence d'élément** avec le type d'invite Sélectionner avec arborescence. Etant donné qu'un type d'invite Sélectionner avec arborescence peut uniquement filtrer sur un niveau ou une hiérarchie, le fait de définir la propriété **Filtrer la référence d'élément** va provoquer une erreur.

### **Test des invites**

Lorsque vous testez un objet de modèle qui fait référence à une invite, IBM Cognos Framework Manager vous demande d'entrer la valeur de l'invite. Framework Manager utilise cette valeur pendant la durée de la session ou jusqu'à ce que vous désélectionniez la valeur de l'invite.

Vous pouvez modifier la valeur de session des valeurs d'invite par le biais de la boîte de dialogue **Options**. Cette boîte de dialogue est accessible lorsque vous modifiez une dimension ou un sujet de requête, ou que vous définissez un calcul, un filtre, un ensemble de requêtes ou une relation complexe. Vous pouvez modifier la valeur d'invite lors du test de l'expression qui fait référence à cette valeur.



Si vous cochez la case **Toujours demander des valeurs lors du test** dans la boîte de dialogue **Invite**, Framework Manager vous invite à indiquer une valeur chaque fois que vous testez l'objet. Lorsque vous mettez à jour l'objet ou que effectuez un comptage, Framework Manager utilise la valeur d'invite existante, le cas échéant.

Une invite sur un élément de requête d'un sujet de requête de modèle est uniquement associée à cet élément de requête. Une invite sur un sujet de requête de source de données est associée à la totalité du sujet de requête. Par conséquent, l'invite s'affiche lorsque vous testez n'importe quel élément de requête du sujet de requête.

## Conversion d'éléments de requête en mesures

Vous pouvez reconverter un élément de requête d'une dimension de mesure en mesure.

Pour convertir une mesure en élément de requête, voir «Conversion de mesures en éléments de requête», à la page 119.

### Procédure

1. Cliquez deux fois sur la dimension de mesure contenant l'élément de requête.
2. Cliquez sur l'onglet **Dimension de mesure**.
3. Cliquez avec le bouton droit de la souris sur l'élément de requête et cliquez sur **Convertir en mesure**.
4. Cliquez sur **OK**.

---

## Règles métier

Vous pouvez ajouter des règles métier aux dimensions et aux sujets de requête de votre modèle pour affiner les données extraites et faire en sorte que vos utilisateurs accèdent à des informations correctes.

Créer des règles métier et les stocker dans le modèle plutôt que dans des rapports présentent de nombreux avantages. Vous gagnez du temps car il n'est plus nécessaire de recréer les règles métier chaque fois qu'elles sont requises. Les règles métier assurent une cohérence car vos utilisateurs utilisent tous les mêmes définitions. Par exemple, la signification de l'expression Marges inférieures est identique dans toute l'organisation. Elles sont faciles à mettre à jour car vous les gérez de manière centralisée. Ainsi, tous les rapports sont mis à jour automatiquement à mesure que les règles évoluent. Par exemple, si la définition de l'expression Marge inférieure change, tous les rapports qui utilisent le calcul Marge inférieure sont mis à jour automatiquement. Les règles métier permettent d'améliorer la sécurité.

Par exemple, vous pouvez :

- Ajouter des calculs de sorte que vos utilisateurs puissent inclure des données calculées dans leurs rapports. Voir «Création de calculs», à la page 152.
- Créer et appliquer des filtres visant à limiter les données extraites par un sujet de requête. Voir «Création de filtres», à la page 155.
- Ajouter des invites qui s'affichent automatiquement chaque fois qu'une dimension ou un sujet de requête est utilisé dans un rapport ; les consommateurs du rapport sont alors invités à filtrer les données. Voir «Définition des contrôles d'invite», à la page 145.

- Utiliser des paramètres de session et des tables de mappage des paramètres dans des macros pour résoudre les expressions de façon dynamique. Voir «Création de paramètres de session», à la page 163, «Création de tables de mappage des paramètres», à la page 160, «Création d'invites avec des macros de requête», à la page 165.
- Créer un filtre de sécurité pour contrôler les données affichées pour vos utilisateurs lorsqu'ils configurent leurs rapports. Voir «Ajout de sécurité des données», à la page 261.

## Création de calculs

Vous pouvez créer des calculs afin de fournir à vos utilisateurs des valeurs calculées qu'ils utiliseront régulièrement. Les calculs peuvent utiliser des éléments de requête, des paramètres, des variables, des membres calculés, des expressions et des composants d'expression, tels que des fonctions.

Les caractères de ponctuation, tels que le point d'interrogation (?), doivent être exprimés en code de caractère ASCII 7 bits. Si vous tapez un caractère de ponctuation à l'aide d'un clavier multi-octets activé, prenez soin d'entrer la représentation ASCII 7 bits du caractère. Par exemple, tapez Alt+063 pour le point d'interrogation.

Évitez de nommer le calcul à l'aide de caractères utilisés pour les opérateurs d'expression. Des erreurs de syntaxe sont susceptibles de se produire lorsque l'expression est évaluée. Par exemple, un calcul nommé `Margin * 10` génère des erreurs lorsqu'il est utilisé dans une expression telle que `[Margin * 10] < 20`.

Dans les expressions, un type dimensionnel spécifique peut être requis pour les opérandes d'un opérateur ou d'une fonction. Lorsqu'un opérande n'est pas du type requis, une ou plusieurs règles de conversion peuvent être appliquées pour convertir l'opérande au type approprié. Étant donné que les règles de conversion ne sont pas appliquées aux expressions des sujets de requête de modèle, vérifiez que ces expressions sont valides sans vous baser sur les règles de conversion. Pour plus d'informations sur les règles de conversion, voir le document IBM Cognos Report Studio *User Guide*.

Si vous insérez dans le calcul une fonction définie par l'utilisateur importée, assurez-vous que le nom de la fonction ne reprend pas le nom d'un fournisseur. Par exemple, si le nom de la fonction définie par l'utilisateur est CHAR, une erreur est générée lorsque vous testez la fonction via la boîte de dialogue **Définition du calcul** car ce nom est considéré comme étant identique à `char` dans Microsoft SQL Server. Pour plus d'informations sur les noms de fonction utilisés dans votre base de données, voir la documentation produit de la base de données.

Lors de la requête, IBM Cognos Framework Manager renvoie une valeur nulle pour les calculs qui contiennent un diviseur dont la valeur est égale à zéro. Framework Manager ne parvient pas à détecter les erreurs de division par zéro dans des fonctions telles que `average` et `mod`, car l'opérateur de division n'est pas explicite.

Framework Manager prend en charge les calculs autonomes et les calculs imbriqués.

- Utilisez un calcul autonome lorsque vous souhaitez réutiliser l'expression ou commander l'ordre des opérations à l'aide du paramètre Calculé de la propriété Agrégat ordinaire. Pour plus d'informations, voir «Définition de l'ordre des opérations pour les calculs de modèles», à la page 344.

Vous pouvez appliquer un calcul autonome à un ou plusieurs sujets de requête ou dimensions afin de fournir des données calculées à un rapport, ou inclure ce calcul dans un pack afin de le rendre disponible pour vos utilisateurs. Le fait de déplacer un calcul autonome ou un raccourci vers celui-ci dans un dossier vous permet d'organiser les objets du modèle de façon plus efficace.

Il est impossible d'utiliser des calculs autonomes dans Analysis Studio. A la place, vous pouvez utiliser un calcul imbriqué dans la dimension de mesure.

- Utilisez un calcul imbriqué lorsque vous souhaitez utiliser un calcul avec une seule dimension ou un seul sujet de requête.

Vous pouvez créer un calcul imbriqué lors de la modification d'un sujet de requête de source de données relationnelle, d'un sujet de requête de modèle ou d'une dimension.

Si vous démarrez à l'aide d'un calcul imbriqué, vous pouvez ensuite le convertir en une expression autonome que vous pouvez appliquer à d'autres dimensions ou sujets de requête. **Astuce** : cliquez avec le bouton droit de la souris sur l'expression de calcul sous l'onglet **Calculs**, puis cliquez sur **Convertir en calcul autonome**.

Lorsque vous imbriquez un calcul, le sujet de requête de source de données doit avoir une relation avec n'importe quel sujet de requête référencé par l'expression. Cette relation est nécessaire même si l'expression fait référence à un sujet de requête de modèle basé sur la même table que le sujet de requête de source de données dans lequel vous incorporez l'expression.

## Procédure de création d'un calcul


### Procédure


1. Effectuez l'une des opérations suivantes :
  - Pour créer un calcul autonome, cliquez sur l'espace-noms ou le dossier, puis sur **Actions, Créer, Calcul**.
  - Pour créer un calcul intégré pour une dimension de mesure, cliquez deux fois sur la dimension. Dans l'onglet **Dimension de mesure**, cliquez sur **Ajouter**.
  - Pour créer un calcul intégré pour une dimension ordinaire, cliquez deux fois sur la dimension. Dans l'onglet **Dimension**, sélectionnez une hiérarchie et cliquez sur **Ajouter**.
  - Pour créer un calcul intégré pour un sujet de requête de source de données, cliquez deux fois sur le sujet de requête de source de données. Dans l'onglet **Calculs**, cliquez sur **Ajouter**.
  - Pour créer un calcul intégré pour un sujet de requête de modèle, cliquez deux fois sur le sujet de requête de modèle. Dans l'onglet **Définition de sujet de requête**, cliquez sur **Ajouter**.

La boîte de dialogue **Définition du calcul** s'affiche.

2. Dans la zone **Nom**, saisissez un nom à attribuer au calcul.
3. Définissez l'expression.

Objectif	Action
Ajouter des éléments	Dans l'onglet <b>Modèle</b> , cliquez sur un élément de requête, un filtre, ou un calcul, puis cliquez sur la flèche.

Objectif	Action
Ajouter des fonctions	Dans l'onglet <b>Fonctions</b> , sélectionnez un composant et cliquez sur la flèche.
Ajouter des paramètres	Dans l'onglet <b>Paramètres</b> , sélectionnez un paramètre et cliquez sur la flèche.
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cliquez sur le bouton d'options, cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes à renvoyer.</p>  <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>
Redéfinir des paramètres de session	Cliquez sur le bouton d'options, puis sur <b>Définir</b> , entrez une valeur dans la zone <b>Remplacer la valeur</b> et cliquez sur <b>OK</b> .
Redéfinir des valeurs d'invite	<p>Cliquez sur le bouton d'options, puis sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>

4. Pour tester le calcul, cliquez sur le bouton de **test**  .  
 Vous pouvez uniquement tester des calculs qui contiennent des éléments de requête. Si un calcul contient une fonction, par exemple, `_add_days`, le bouton **Tester l'exemple** n'est pas disponible.  
**Astuce** : Si le calcul comprend une expression non valide, consultez la zone **Conseils** dans l'éditeur d'expression pour plus d'informations.
5. Cliquez sur **OK**.
6. Modifiez la propriété **Type de données** pour identifier le type de donnée renvoyé par le calcul.  
 Le studio IBM Cognos utilise ces informations pour formater les données renvoyées par le calcul.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Annexe C, «Utilisation de l'éditeur d'expression», à la page 403
- «Création de paramètres de session», à la page 163
- «Test de dimensions», à la page 121

## Création de filtres

Un filtre est une expression qui définit les conditions que les lignes ou les instances doivent satisfaire pour être extraites pour la dimension, le sujet de requête, le calcul ou le rapport auquel le filtre est appliqué. Un filtre renvoie une valeur booléenne, ce qui vous permet de limiter les lignes renvoyées par une dimension ou un sujet de requête.

Par exemple, vous pouvez utiliser la fonction `in_range` pour créer un filtre qui extrait les données pour les produits introduits dans un délai imparti déterminé. La syntaxe de cet exemple se présente comme suit :

```
[gosales_goretailers].[Products].[Introduction  
date]  
in_range {Feb 14, 1999 : July 14, 2007}
```

**Remarque :** lorsque vous utilisez une fonction de date ou d'heure, vous devez utiliser une horloge au format 24 heures. IBM Cognos Framework Manager ne prend pas en charge les mentions "a.m." et "p.m." dans les expressions. Par exemple, vous utiliserez 20:00 pour indiquer 8 p.m.

Vous pouvez limiter les données représentées par les dimensions ou les sujets de requête dans un projet en créant un filtre de sécurité. Le filtre de sécurité permet de contrôler les données visibles par vos utilisateurs lorsqu'ils configurent leurs rapports.

Vous pouvez également appliquer des restrictions afin de limiter les données extraites par les requêtes d'un pack.

Framework Manager prend en charge les filtres autonomes et les filtres imbriqués.

- Utilisez un filtre autonome lorsque vous souhaitez réutiliser l'expression.  
Vous pouvez ajouter un filtre autonome à un ou plusieurs sujets de requête ou dimensions afin de limiter les données extraites par la requête lorsque le sujet de requête ou la dimension filtrés sont utilisés dans un rapport, ou vous pouvez inclure ce filtre autonome dans un pack afin de le rendre disponible pour vos utilisateurs. Le fait de déplacer un filtre autonome ou un raccourci vers celui-ci dans un dossier vous permet d'organiser les objets du modèle de façon plus efficace.

- Utilisez un filtre imbriqué lorsque vous souhaitez utiliser un filtre avec une seule dimension ou un seul sujet de requête.

Vous pouvez créer un filtre imbriqué lors de la modification d'une dimension, d'un sujet de requête de source de données relationnelle ou d'un sujet de requête de modèle.

Si vous démarrez à l'aide d'un filtre imbriqué, vous pouvez ensuite le convertir en une expression autonome que vous pouvez appliquer à d'autres dimensions ou sujets de requête. **Astuce :** cliquez avec le bouton droit de la souris sur l'expression de filtre sous l'onglet **Filtres**, puis cliquez sur **Convertir en filtre autonome**.

Lorsque vous imbriquez un filtre, le sujet de requête de source de données doit avoir une relation avec n'importe quel sujet de requête référencé par l'expression. Cette relation est nécessaire même si l'expression fait référence à un sujet de requête de modèle basé sur la même table que le sujet de requête de source de données dans lequel vous incorporez l'expression.


Pour créer un filtre sur un sujet de requête non associé, procédez de l'une des façons suivantes :

- Vérifiez qu'il existe un chemin de jointure entre le nouveau sujet de requête nouvelle et celui contenant le filtre.
- Fondez le filtre imbriqué sur un élément de requête basé sur le sujet de requête de source de données souhaité.
- Convertissez le filtre en un filtre autonome, de sorte qu'il ne fasse pas partie du sujet de requête.
- Créez un filtre autonome qui fait référence à l'objet imbriqué.

## Procédure de création de filtres

### Procédure

1. Effectuez l'une des opérations suivantes :
  - Pour créer un filtre autonome, cliquez sur l'espace-noms ou le dossier, puis sur **Actions, Créer, Filtre**.
  - Si vous souhaitez créer un filtre intégré, cliquez deux fois sur la dimension ou le sujet de requête et cliquez sur l'onglet **Filtres**, puis sur **Ajouter**.
2. Dans la zone **Nom**, saisissez un nom à attribuer au filtre.
3. Définissez l'expression.  
**Astuce** : Si le filtre comprend une expression non valide, consultez la zone **Conseils** dans l'éditeur d'expression pour plus d'informations.

Objectif	Action
Ajouter des éléments de requête et des filtres	Dans l'onglet <b>Modèle</b> , faites glisser les objets souhaités vers la zone <b>Définition de l'expression</b> .
Ajouter des fonctions	Dans l'onglet <b>Fonctions</b> , faites glisser les fonctions vers la zone <b>Définition de l'expression</b> .
Ajouter des paramètres	Dans l'onglet <b>Paramètres</b> , faites glisser les paramètres vers la zone <b>Définition de l'expression</b> .
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cliquez sur le bouton d'options, cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes à renvoyer.</p>  <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>
Redéfinir des paramètres de session	Cliquez sur le bouton d'options, puis sur <b>Définir</b> , entrez une valeur dans la zone <b>Remplacer la valeur</b> et cliquez sur <b>OK</b> .

Objectif	Action
Redéfinir des valeurs d'invite	<p>Cliquez sur le bouton d'options, puis sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>

4. Cliquez sur **OK**.

## Résultats

Vous pouvez également appliquer des restrictions afin de limiter les données extraites par les requêtes d'un pack. Pour plus d'informations, voir «Restrictions», à la page 309.

Vous pouvez être amené à consulter les rubriques associées suivantes :

- «Ajout de sécurité des données», à la page 261
- Annexe C, «Utilisation de l'éditeur d'expression», à la page 403
- «Création de tables de mappage des paramètres», à la page 160
- «Création de paramètres de session», à la page 163
- «Test de sujets de requête ou d'ensembles de requêtes», à la page 93
- «Paramètres et invites obligatoires», à la page 169

## Application de filtres

Pour appliquer un filtre, vous devez modifier la dimension, le sujet de requête de source de données, ou le sujet de requête de modèle. Le sujet de requête doit contenir les éléments de requête référencés par le filtre, ou disposer d'un chemin de relation vers les sujets de requête qui contiennent les éléments de requête.

Vous pouvez incorporer un filtre autonome dans les dimensions ou les sujets de requête, mais si vous souhaitez une utilisation différente pour chaque filtre incorporé, vous devez créer différentes versions du filtre autonome. Sinon, il se peut que vos utilisateurs soient obligés de répondre à une invite que vous pensiez être facultative s'il existe une instance dans laquelle l'utilisation a été définie comme obligatoire. Pour plus d'informations sur les invites facultatives et obligatoires, voir «Paramètres et invites obligatoires», à la page 169.

Par exemple, dans le sujet de requête A, vous incorporez un filtre autonome que vous définissez comme étant facultatif. Dans le sujet de requête B, vous le définissez comme étant obligatoire. Lorsque vos utilisateurs créent un rapport qui utilise les deux sujets de requête, il leur est demandé de choisir des valeurs dans les deux filtres, même dans celui qui est défini comme étant facultatif. Toutes les instances du filtre sont considérées comme étant obligatoires lorsqu'elles sont utilisées dans la même requête. La solution consiste à créer différentes versions du filtre, chacun ayant un nom distinct.

## Procédure d'application d'un filtre

### Procédure

1. Créez un filtre.
2. Sélectionnez le filtre, puis cliquez sur **Actions, Editer la définition**.
3. Cliquez sur l'onglet **Filtres**, puis faites glisser le filtre que vous avez créé vers la zone **Filtres**.
4. Sélectionnez une valeur d'utilisation pour votre filtre.

Valeur d'utilisation	Description
Toujours	<p>Utilisez cette valeur d'utilisation pour que les données spécifiées soient filtrées dans les rapports. Par exemple, votre entreprise peut stocker des informations obsolètes sans vouloir générer des rapports sur celles-ci.</p> <p><b>Toujours</b> est la valeur d'utilisation par défaut.</p>
Mode de conception uniquement	<p>Extrait un petit sous-ensemble des données pour l'exemple de rapport. Utilisez cette valeur d'utilisation lorsque vous n'avez pas besoin de voir toutes les données, par exemple, lorsque vous testez un sujet de requête.</p> <p>Pour appliquer des filtres de mode de conception dans Framework Manager, sélectionnez l'option <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b>. Cette option est disponible dans l'onglet <b>Tester les paramètres</b>.</p> <p>Les utilisateurs peuvent avoir besoin du filtrage de mode de conception dans Query Studio quand ils veulent se concentrer sur la conception de la présentation et du format d'un rapport, et ne pas extraire toutes les données quand ils travaillent. Pour accéder au filtrage de mode de conception dans Query Studio, exécutez le rapport avec un sous-ensemble de données.</p>



Valeur d'utilisation	Description
Facultatif	<p>Indique si le filtre est facultatif. L'utilisateur est invité à filtrer les données et peut laisser la zone vide. Si l'invite est vide, Framework Manager ignore le filtre et extrait toutes les données de la dimension ou du sujet de requête.</p> <p>La syntaxe ?? est requise pour les invites facultatives.</p> <p>Faites appel à cette valeur d'utilisation si les utilisateurs veulent contrôler à quel moment appliquer le filtre. Par exemple, vous souhaitez afficher parfois un pays et d'autres fois visualiser les données pour tous les pays. Un filtre facultatif pour les pays se présente comme suit :</p> <pre> [[GeoNamespace].[Countries] .[CountryName] = ?WhichCountry?) </pre>

- Si vous souhaitez afficher le code SQL, cliquez sur l'onglet **Informations sur la requête**.
- Cliquez sur **OK**.

## Exemple - Affichage du nom de la devise pour chaque pays

Vous souhaitez créer une requête qui affiche le nom de la devise pour un pays spécifique. Pour ce faire, vous créez un filtre qui renvoie des données pour un code pays spécifique et vous appliquez le filtre à un sujet de requête de modèle qui extrait le nom de la devise pour chaque pays.

L'exemple suivant utilise une source de données relationnelles.

### Procédure

- Ouvrez l'exemple de modèle ventesVA. Il est situé dans `c10_location/webcontent/samples/Models/go_sales/go_sales.cpf`
- Créez un filtre pour limiter l'extraction des données aux codes de pays de la table de taux de conversion dont la valeur est 2 :
  - Cliquez sur le dossier **Filtres**, puis cliquez sur **Actions, Créer, Filtre** et nommez le nouveau filtre **CodePaysTauxConversion**.
  - Cliquez sur l'onglet **Modèle**.
  - Dans la zone **Composants disponibles**, ouvrez le dossier de vue Base de données, puis ouvrez le dossier GoSales.
  - Ajoutez l'élément de requête Code de pays du sujet de requête Taux de conversion à la zone **Définition de l'expression**, puis entrez `=2` à la fin de l'expression.
  - Cliquez sur **OK**.
- Créez un sujet de requête de modèle nommé **Code ISO**.
  - Dans la zone **Objets de modèle disponibles**, ouvrez le dossier de vue Base de données.

- Ajouter l'élément de requête Pays et l'élément de requête ISO 3 lettres du sujet de requête Pays à la zone **Eléments de requête et calculs**.
4. Appliquez le filtre CodePaysTauxConversion :
    - Cliquez sur l'onglet **Filtres**.
    - Ouvrez le dossier Filtres et faites glisser CodePaysTauxConversion vers la zone **Filtres**.
  5. Cliquez sur l'onglet **Informations sur la requête**.  
Le code SQL généré contient le filtre, même s'il n'a pas d'incidence sur l'ensemble de résultats.
  6. Modifiez l'utilisation du filtre CodePaysTauxConversion en **Facultatif** :
    - Cliquez sur l'onglet **Filtres**.
    - Cliquez sur les points de suspension (...) sous **Utilisation** pour le filtre CodePaysTauxConversion, puis cliquez sur **Facultatif**.
  7. Si vous souhaitez afficher le code SQL, cliquez sur l'onglet **Informations sur la requête**.
  8. Cliquez sur **OK**.

## Création de tables de mappage des paramètres

Utilisez des paramètres pour créer des sujets de requête conditionnels permettant d'activer des substitutions lors de l'exécution du rapport. Les tables de mappage des paramètres sont des objets qui stockent des paires clé-valeur.

Les tables de mappage des paramètres sont semblables aux tables de recherche de source de données. Chaque table de mappage des paramètres comporte deux colonnes, l'une contenant la clé et l'autre contenant la valeur représentée par la clé. Vous pouvez entrer manuellement les clés et les valeurs, les importer à partir d'un fichier ou les baser sur des éléments de requête du modèle.

Vous pouvez également exporter des tables de mappage des paramètres dans un fichier. Pour modifier la table de mappage des paramètres, vous pouvez exporter les valeurs qu'elle contient dans un fichier, effectuer des ajouts ou des modifications, puis l'importer à nouveau dans IBM Cognos Framework Manager. Cela s'avère particulièrement utile lorsque vous utilisez des tables de mappage des paramètres volumineuses et complexes.

Toutes les clés de table de mappage des paramètres doivent être uniques pour que Framework Manager puisse extraire la valeur correcte de façon cohérente. Vous ne devez pas placer la valeur de paramètre entre guillemets. Vous pouvez utiliser des guillemets dans l'expression dans laquelle le paramètre est utilisé.

La valeur d'un paramètre peut être un autre paramètre. Toutefois, vous devez placer la totalité de la valeur entre signes dièses (#). L'imbrication de paramètres en tant que valeurs est limitée à cinq niveaux.

Lorsque vous utilisez une table de mappage des paramètres en tant qu'argument sur une fonction, vous devez utiliser le symbole du pourcentage (%) au lieu du symbole du dollar (\$).

Attribuez un alias à un élément de requête dont le nom se compose en partie d'une table de mappage des paramètres dans son nom et pour ajouter des noms multilingues à l'objet dans l'onglet **Langue** (sous-fenêtre **Propriétés**).

Vous ne devez pas baser une table de mappage des paramètres sur un élément de requête ou une table avec un ensemble de résultats volumineux, par exemple 50 000 lignes. Chaque fois que vous utilisez la table de mappage des paramètres dans une expression ou dans SQL, Framework Manager exécute cette requête volumineuse. Les performances s'en trouvent ralenties. Les tables de mappage des paramètres doivent être utilisées pour des tables de recherche plus petites.

## Procédure de création manuelle d'une table de mappage des paramètres

### Procédure

1. Cliquez sur le dossier **Tables de mappage de paramètres**, puis sur **Actions, Créer, Table de mappage de paramètres**.
2. Dans la zone **Nom**, saisissez un nom à attribuer à la nouvelle table de mappage des paramètres.
3. Cliquez sur **Entrez manuellement les clés de paramètre et/ou importez-les à partir d'un fichier**, puis sur **Suivant**.
4. Effectuez l'une des opérations suivantes :
  - Pour entrer manuellement des valeurs, cliquez sur **Nouvelle clé**, entrez une clé et appuyez sur la touche de tabulation pour saisir une valeur pour cette clé.
  - Pour importer des clés et des valeurs, cliquez sur **Importer un fichier** et identifiez l'emplacement du fichier .csv ou .txt approprié. Pour un fichier .txt à utiliser pour l'importation, les valeurs doivent être séparées par des tabulations et le fichier doit être enregistré au format UTF8 ou Unicode. Les fichiers texte ANSI ne sont pas pris en charge.

**Remarque :** Si vous prévoyez d'utiliser un paramètre dans un sujet de requête de source de données, la valeur doit utiliser la ponctuation anglaise. Cela signifie que vous devez utiliser un point (.) pour représenter une décimale et une virgule (,) pour séparer des valeurs dans les listes.
5. Modifiez les paramètres existants selon les besoins.

Objectif	Action
Affecter une valeur par défaut	<p>Dans la zone <b>Valeur par défaut</b>, saisissez une valeur.</p> <p>Si la clé utilisée dans une expression n'est pas mappée, la valeur par défaut est utilisée.</p> <p>La définition d'une valeur par défaut est facultative. Cependant, si aucune valeur par défaut n'est fournie, une clé non mappée risque de générer une erreur.</p>
Charger à la demande	<p>Le chargement d'une table de mappage des paramètres contenant un grand nombre de clés peut être long lors de l'exécution d'un rapport. Vous pouvez limiter le chargement aux clés qui sont requises pour un rapport spécifique en définissant <b>Charger à la demande</b> sur true.</p> <p>Par défaut, la valeur est définie sur false et toutes les clés sont chargées.</p>

Objectif	Action
Supprimer un paramètre	Sélectionnez une ligne et cliquez sur <b>Supprimer</b> .
Modifier un paramètre	Sélectionnez la ligne à modifier, cliquez sur le bouton <b>Editer</b> et entrez une valeur.
Effacer toutes les clés et les valeurs	Cliquez sur <b>Clear Map</b> (Effacer la mappe).

6. Cliquez sur **Terminer**.

## Procédure de création d'une table de mappage des paramètres à partir d'éléments de requête existants

### Procédure

1. Cliquez sur le dossier **Tables de mappage de paramètres**, puis sur **Actions, Créer, Table de mappage de paramètres**.
2. Dans la zone **Nom**, saisissez un nom à attribuer à la nouvelle table de mappage des paramètres.
3. Cliquez sur **Définir la table de mappage des paramètres en fonction des éléments de requête, Suivant**.
4. Cliquez sur l'élément de requête à utiliser comme clé, puis cliquez sur l'élément de requête devant servir de valeur.  
Les éléments de requête doivent provenir du même sujet de requête.
5. Cliquez sur **Suivant**.
6. Dans la zone **Valeur par défaut**, saisissez une valeur.  
Si la clé utilisée dans une expression n'est pas mappée, la valeur par défaut est utilisée.  
La définition d'une valeur par défaut est facultative. Cependant, si aucune valeur par défaut n'est fournie, une clé non mappée risque de générer une erreur.
7. Cliquez sur **Terminer**.

### Résultats

## Exemple - Spécification de valeurs de langue pour des métadonnées relationnelles

Une entreprise de niveau international stocke ses informations produit en anglais et en français. A l'aide d'une table de mappage des paramètres et de macros, les employés peuvent extraire des données qui correspondent aux informations dont ils ont besoin.

Créez une table de mappage des paramètres Language\_lookup contenant les données suivantes :

Clé	Valeur
fr	FR
fr	FR

Lorsque vous examinez les instructions SQL pour le sujet de requête Ligne de produits, vous pouvez voir les rubriques suivantes :

```
Select
  PRODUCT_LINE.PRODUCT_LINE_CODE,
  #'PRODUCT_LINE.PRODUCT_LINE_' + $Language_lookup{$runLocale}#
as Product_Line
from
  [gosa]es].PRODUCT_LINE PRODUCT_LINE
```

La macro runLocale renvoie des paramètres régionaux qui sont utilisés par la macro Language\_lookup pour spécifier une valeur de langue.

## Création de paramètres de session

Un paramètre de session est une variable qu'IBM Cognos Framework Manager associe à une session.

Par exemple, l'ID utilisateur et la langue préférée sont deux paramètres de session. Les paramètres de session étant des paires clé-valeur, vous pouvez envisager chaque paramètre de session comme une entrée d'une table de mappage des paramètres appelée Paramètres de session. Vous pouvez utiliser un paramètre de session de la même façon que vous utilisez une entrée de table de mappage des paramètres, même si la syntaxe pour les paramètres de session est légèrement différente.

Il existe deux types de paramètre de session : environnement et modèle.

les paramètres de session d'environnement sont prédéfinis et stockés dans Content Manager. Par défaut, les paramètres de session suivants s'affichent dans Framework Manager :

- **runLocale**

Renvoie le code de la langue active en cours dans Framework Manager. Le contenu de modèle est affiché dans cette langue.

Vous pouvez modifier la langue active à tout moment pour votre session en cours uniquement. Dans les sessions futures, le modèle continuera de s'ouvrir dans la langue utilisée lors de la conception. Pour plus d'informations, voir la section «Ajout d'une langue à un projet», à la page 129.

- **account.defaultName**

Indique le nom de l'utilisateur en cours tel qu'il est défini dans le fournisseur d'authentification. Par exemple, le prénom et le nom de l'utilisateur.

Si vous vous connectez en tant qu'anonyme, **Anonyme** est affiché.

- **account.personalInfo.userName**

Indique l'ID utilisateur sous lequel la connexion à IBM Cognos BI est établie.

Si vous vous connectez en tant qu'anonyme, ce paramètre n'est pas affiché.

- **current\_timestamp**

Indique la date et l'heure en cours.

- **machine**

Indique le nom de l'ordinateur sur lequel Framework Manager est installé.

Si votre source d'authentification prend en charge d'autres paramètres et que vous avez entré des informations les concernant dans la source d'authentification, d'autres paramètres de session sont affichés, tels que account.personalInfo.email ou account.personalInfo.surname.

Vous pouvez définir d'autres paramètres en utilisant les paramètres de session de modèle. Les paramètres de session de modèle sont stockés dans une table de mappage des paramètres appelée `_env`. Ils sont définis dans le projet et peuvent être publiés avec un pack.

Les valeurs définies pour les paramètres de session de modèle doivent être comprises dans la portée des objets du modèle Framework Manager. Cette portée peut inclure l'utilisation des paramètres de session d'environnement existants, ainsi que des valeurs statiques.

Chaque paramètre de session doit porter un nom et posséder une valeur par défaut. Vous pouvez définir une valeur de substitution pour tester les résultats renvoyés par cette valeur. La valeur de substitution est valide uniquement lorsque le modèle est ouvert ; elle n'est pas enregistrée lors de la sauvegarde du modèle. En l'absence de valeur de substitution, Framework Manager utilise la valeur par défaut lorsqu'il exécute une requête qui contient un paramètre de session.

Les règles qui régissent l'utilisation de paramètres sont les suivantes :

- Toutes les valeurs de retour possibles doivent posséder le même type de données.
- Une seule valeur peut être définie.

## Procédure de création de paramètres de session

### Procédure

1. Cliquez sur **Projet, Paramètres de session**.
2. Cliquez sur **Nouvelle clé** et entrez une clé et une valeur de paramètre de session.
3. Sélectionnez le mode de traitement de la valeur de remplacement.
  - Afin d'éviter d'avoir à définir la valeur de remplacement chaque fois que vous éditez le projet, définissez le paramètre de session en tant que valeur.
  - Afin d'éviter d'avoir à supprimer le paramètre de projet chaque fois que vous le publiez, définissez le paramètre de session en tant que remplacement de session.
4. Modifiez les paramètres existants selon les besoins.

Objectif	Action
Modifier la valeur de paramètre	Cliquez sur la ligne qui contient la valeur à modifier, puis sur <b>Editer</b> et entrez une valeur.
Affecter une valeur par défaut	Dans la zone <b>Valeur par défaut</b> , saisissez une valeur. Framework Manager utilise la valeur par défaut si une clé a une valeur non valide.
Supprimer un paramètre	Cliquez sur une ligne, puis sur le bouton <b>Supprimer</b> . Vous ne pouvez pas supprimer un paramètre de session d'environnement.
Effacer une valeur de remplacement	Cliquez sur une ligne, puis sur <b>Effacer le remplacement</b> .

5. Cliquez sur **OK**.

## Utilisation de paramètres avec des sujets de requête de source de données relationnelles

Les objets de modèle ne reflètent pas les modifications apportées aux objets de source de données sur lesquels ils sont basés. Par conséquent, lorsque vous ajoutez un paramètre à un sujet de requête de source de données, il peut être judicieux de créer un objet de modèle faisant référence au paramètre. Dans ce cas, vous devez affecter un alias d'objet paramétré dans le sujet de requête de source de données. Cela garantit que les sujets de requête de modèle, les filtres ou les calculs qui font référence à l'objet renvoient les résultats corrects lorsque la valeur du paramètre est modifiée.

Par exemple, le code SQL suivant définit le sujet de requête de source de données qui contient un paramètre de session appelé `runLocale`. La valeur du paramètre `runLocale` indique la colonne extraite par la requête. L'alias se comporte comme un raccourci pour qu'IBM Cognos Framework Manager extraie la valeur à laquelle l'alias est affecté lorsqu'un objet de modèle fait référence à `CountryNameAlias`.

```
Select
#ColumnMap{$runLocale}# as
CountryNameAlias
From
[GoSales].Country
```

## Création d'invites avec des macros de requête

Les macros sont des fragments de code que vous pouvez insérer à n'importe quel endroit de l'instruction `Select` qui définit un sujet de requête. Vous pouvez inclure des références à des paramètres de session, des tables de mappage de paramètres et des entrées de table de mappage de paramètres. Les valeurs des paramètres sont définies lorsque vous exécutez la requête.

Par exemple, vous pouvez utiliser le paramètre de session de langue pour n'afficher que les données qui correspondent au paramètre linguistique de l'utilisateur en cours.

Les macros peuvent être utilisées de différentes manières :

- Elles peuvent être insérées dans le code SQL.  
Exemple : `Select * from Country where Country.Name = #myMap{$runLocale}#`
- Elles peuvent fournir un argument à un sujet de requête de procédure stockée.  
Si une valeur n'est pas codée en dur pour l'argument, le sujet de requête de procédure stockée peut être utilisé pour renvoyer des données différentes.
- Elles peuvent être insérées dans des expressions, telles que des calculs et des filtres.  
Exemple de filtre : `[gosales].[Sales staff].[Staff name] = #UserLookUpMap{$UserId}#`
- Elles peuvent être utilisées pour exécuter dynamiquement les propriétés d'un sujet de requête de source de données.  
Cela permet aux différents utilisateurs d'entrer des informations de connexion différentes et d'accéder ainsi à des sources de données différentes. Les propriétés pouvant contenir des macros sont les suivantes : **Source de données de Content Manager**, **Catalogue**, **Cube** et **Schéma**.  
Voici un exemple utilisant la propriété **Source de données de Content Manager** : `#DataSourceMap{$UserId}#`
- Elles peuvent être utilisées sous la forme d'un assistant de paramètres.

Des paramètres peuvent faire référence à d'autres paramètres. Exemple : Map1, Key = en-us, Value = #myMap{\$UserId}#

- Elles peuvent être utilisées dans la boîte de dialogue **Paramètres de session**.  
Exemple : MySessionParameter, value = #myMap{\$UserGroup}#

Vous pouvez remplacer les éléments de sujet de requête suivants par paramètre.

Elément	Exemple
Éléments de requête identifiés dans la liste Select	#'Product_name_' + \$languageCode#
Tables identifiées dans la clause From	Product_#\$languageCode#
Clause Where	Product_lang = #sq(\$languageCode)#
Nom de la propriété de source de données, de schéma ou de source	#\$data_source#.\$\$schema#.Products

## Syntaxe

Utilisez la syntaxe suivante pour faire référence à des paramètres et des valeurs de paramètre de session.

Objet	Syntaxe	Exemple
Clé de session	\$session_key	#my_account#
Clé de table de mappage des paramètres	\$map{<key>}	#map_one{'abc'}#
Entrée de table de mappage des paramètres dont la clé est définie par un paramètre de session	\$map{\$session_key}	#map_one{my_account}#

Vous pouvez ajouter les éléments suivants pour affiner la définition de l'expression de macro.

Symbole	Fonction
Apostrophes '	<p>Délimite une chaîne littérale qui comprend une apostrophe.</p> <p>Si l'apostrophe s'affiche dans une chaîne, telle qu'un élément de requête, la chaîne doit être placée entre apostrophes et l'apostrophe à l'intérieur de la chaîne doit être doublée. Par exemple, ab'c apparaît sous la forme 'ab' 'c'</p> <p>Si l'apostrophe s'affiche dans une macro, la chaîne doit être placée entre crochets. Par exemple, ab'c apparaît sous la forme [ab'c]</p> <p>Si l'apostrophe s'affiche dans une invite, il n'est pas nécessaire de délimiter la chaîne.</p> <p>Pour mettre en échappement une apostrophe dans une expression, utilisez &amp;apos;</p>



Symbole	Fonction
Crochets [ ]	Délimite des objets de modèle, tels qu'un espace-noms ou un sujet de requête, et des noms de macro qui contiennent des caractères restreints, comme un signe dièse, un tiret, ou un espace.
Accolades { }	Appelle une fonction qui n'est pas connue par l'analyseur syntaxique, par exemple <code>dateadd</code> dans DB2, et dont le premier argument est un mot clé.  Exemple : <code>dateadd</code> <code>{month},2,&lt;date</code> <code>expression&gt;</code>
Opérateur +	Concatène deux chaînes, par exemple <code>'abc'</code> + <code>'xyz'</code>
Fonction de guillemet simple (sq)	Place le résultat d'une expression de chaîne entre apostrophes. Si l'apostrophe s'affiche dans une chaîne, telle qu'un élément de requête, la chaîne doit être placée entre apostrophes et l'apostrophe à l'intérieur de la chaîne doit être doublée. Vous pouvez utiliser cette fonction pour générer des clauses afin effectuer des tests sur des valeurs littérales reposant sur des paramètres.  Voici un exemple : <code>#sq(\$my_sp)#</code>  Si un paramètre de session ( <code>my_sp</code> ) a la valeur <code>ab'cc</code> , le résultat est <code>'ab"cc'</code>
Fonction de guillemets doubles (dq)	Place le résultat d'une expression de chaîne entre guillemets. Vous pouvez utiliser cette fonction pour faire référence à des noms de table et de colonne avec des caractères non-identificateurs, tel qu'un espace ou un signe de pourcentage (%).  Voici un exemple : <code>#dq</code> <code>('Column' + \$runLocale)#</code>  Si <code>runLocale=en-us</code> , le résultat est <code>"Column en-us"</code>

Symbole	Fonction
Fonction de crochet (sb)	<p>Insère un ensemble de crochets dans l'argument pour créer des références d'objet dans une requête de modèle et des expressions de modèle, telles que des filtres et des calculs.</p> <p>Voici un exemple :</p> <pre>#sb ('my item in ' + \$runLocale)#</pre> <p>Si runLocale=en-us, le résultat est [my item in en-us]</p>

Pour en savoir davantage sur les fonctions, reportez-vous à la section Annexe C, «Utilisation de l'éditeur d'expression», à la page 403.

### Création de macros

Les macros sont des fragments de code que vous pouvez insérer à n'importe quel endroit de l'instruction `Select` qui définit un sujet de requête. Vous pouvez inclure des références à des paramètres de session, des tables de mappage de paramètres et des entrées de table de mappage de paramètres. Les valeurs des paramètres sont définies lorsque vous exécutez la requête.


Lorsque faites référence à un paramètre, vous devez procéder comme suit :

- Utilisez un signe dièse (#) au début et à la fin de chaque ensemble d'un ou plusieurs paramètres.  
Tout ce qui se trouve entre ces deux signes est considéré comme une expression de macro et est traité lors de l'exécution. Framework Manager supprime tous les éléments figurant en dehors de signes dièse.
- Faites précéder chaque entrée de table de mappage des paramètres par le symbole du dollar (\$)
- Utilisez un nom qui commence par un caractère alphabétique (a..z, A..Z)

N'insérez pas de macros entre des guillemets, des apostrophes ou des crochets existants car IBM Cognos Framework Manager n'exécute aucune instruction entre ces éléments.

### Procédure

1. Sélectionnez le sujet de requête de source de données à modifier.
2. Cliquez sur **Actions, Editer la définition**.
3. Dans l'onglet **SQL**, cliquez sur **Insérer une macro** pour lancer l'éditeur de macro.
4. Dans la zone **Composants disponibles**, cliquez sur les tables de mappage des paramètres, les paramètres de session ou les fonctions que vous souhaitez utiliser, puis faites-les glisser vers la zone **Définition de macro**.  
Veillez à entrer l'expression de macro entre les signes dièse. Si vous entrez un texte avant ou après le signe dièse, Framework Manager supprime ce texte lorsque vous cliquez sur OK.
5. Insérez des fonctions d'apostrophe ou de guillemet.

**Astuce :** Cliquez sur la flèche en regard de ces boutons  pour afficher un menu d'options permettant de placer les guillemets ou les apostrophes.

6. Si vous souhaitez éditer une table de mappage des paramètres ou un paramètre de session, cliquez dessus dans la zone **Définition de macro**.

La boîte de dialogue **Table de mappage des paramètres** ou **Paramètres de session** s'affiche. Vous pouvez remplacer des valeurs pour des paramètres de session, ajouter de nouveaux éléments ou modifier des valeurs.

7. Vérifiez la macro dans la zone **Information**.

Si une macro est incorrecte, un message d'erreur s'affiche.

**Astuce :** Pour effacer une macro, cliquez sur l'icône **Tout effacer** .

8. Cliquez sur **OK**.

## Paramètres et invites obligatoires

Vous pouvez créer des invites obligatoires et facultatives dans les modèles IBM Cognos Framework Manager à l'aide de macros de requêtes. Vous pouvez utiliser deux fonctions macros d'invite, `prompt` et `promptmany`, pour créer des invites de choix d'une valeur unique et des invites de choix de plusieurs valeurs. Vous pouvez insérer une macro d'invite à n'importe quel endroit dans l'instruction SQL qui définit le sujet de requête.

Si vous souhaitez utiliser une macro d'invite dans une expression, par exemple, un calcul, vous devez indiquer le type de données lors de l'utilisation d'un opérateur surchargé, tel qu'un signe plus (+). Vous pouvez utiliser le signe plus pour concaténer deux éléments et ajouter deux éléments.

Si vous souhaitez définir un filtre sur un niveau de dimension et lui faire utiliser la macro `prompt` ou `promptmany`, vous devez fournir le type de données `memberuniquename` et une valeur par défaut. Pour plus d'informations sur l'application de filtres, voir «Application de filtres», à la page 157.

Voici un exemple :

```
members( [MS_gosales].[New Dimension].[PRODUCTLINE].[PRODUCTLINE])
  in ( set( #promptmany('what', 'memberuniquename',
  '[MS_gosales].[PROD1].[PRODUCTLINE].[PRODUCTLINE]
->[a11].[1]')# ) )
```

Voici un exemple d'invite obligatoire :

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#
```

Lorsque les valeurs par défaut sont spécifiées dans la syntaxe d'invites de macro, une erreur peut s'afficher. Utilisez une syntaxe invite telle que `Country = ?Entrez le pays?`.

Les fonctions `prompt` et `promptmany` ont les paramètres suivants. Toutes les valeurs d'argument doivent être indiquées sous forme de chaînes.

## Nom

Ce paramètre obligatoire correspond au nom de l'invite. Il peut également faire référence au nom d'un paramètre sur une page d'invite créée par l'utilisateur, auquel cas cette page s'affiche lorsque le rapport est exécuté au lieu de la page d'invite par défaut qui serait générée par la macro.

## Type de données

Ce paramètre facultatif correspond au type de données de la valeur d'invite. La valeur par défaut est "string". Les valeurs d'invite sont validées. Dans le cas de chaînes, la valeur fournie est placée entre apostrophes et les apostrophes incorporées sont doublées.

Les valeurs sont les suivantes :

- boolean
- date
- datetime
- decimal
- double
- float
- int
- integer
- interval
- long
- memberuniqueusername

Il ne s'agit pas réellement d'un type de données. Cette valeur doit être utilisée lorsque le type de données du paramètre correspond à un nom unique de membre.

- numeric
- real
- short
- string
- time
- timeinterval
- timestamp
- token

Il ne s'agit pas réellement d'un type de données. Il s'agit d'un moyen de transmettre du code SQL. Un jeton ne transmet pas de valeurs.

- xsddate
- xsddatetime
- xsddecimal
- xsddouble
- xsdduration
- xsdfloat
- xsdint
- xsdlong
- xsdshort

- xsdstring
- xsdtime

### DefaultText

Ce paramètre facultatif correspond au texte à utiliser par défaut. Si une valeur est spécifiée, l'invite est facultative.

Si vous utilisez un espace et qu'aucune valeur n'est fournie dans la boîte de dialogue **Valeur d'invite**, en règle générale, une clause Where n'est pas générée.

Si vous utilisez un texte et qu'aucune valeur n'est fournie dans la boîte de dialogue **Valeur d'invite**, en règle générale, une clause Where est générée à l'aide de la valeur par défaut.

Assurez-vous que le texte que vous saisissez permet de produire une instruction SQL valide.

**Remarque :** si le type de données est memberuniqueName, une valeur doit être fournie pour le paramètre DefaultText. Par exemple :

```
(#prompt('WhichLevel', 'memberuniqueName',
'[goSalesAgain].[PRODUCT1].[PRODUCT].[PRODUCT(A11)]->[a11]')#)
```

### Text

Ce paramètre facultatif correspond au texte qui précède toute valeur fournie par l'utilisateur, telle que 'and column1 = '.

### QueryItem

Ce paramètre est facultatif. Le moteur d'invite peut tirer parti des propriétés **Informations d'invite** de l'élément de requête. Il est possible d'afficher des informations descriptives, même si la valeur d'invite est un code.

### TextFollowing

Ce paramètre facultatif est la parenthèse de fermeture utilisée le plus souvent avec la fonction promptmany. Ce paramètre est également utile lorsque l'invite est facultative et qu'elle est suivie de filtres codés en dur dans l'instruction SQL.

### Exemples : Sélection des invites de pays ou de région

Lorsqu'un rapport est exécuté, vous souhaitez que vos utilisateurs soient invités à choisir le pays ou la région pour lesquels ils souhaitent afficher des données. Les exemples de code ci-après illustrent la façon dont vous pouvez utiliser des macros pour créer différentes invites.

#### Invite obligatoire sans type de données spécifié :

Tenez compte des points suivants :

- L'argument Datatype n'est pas spécifié. Par conséquent, il s'agit d'une chaîne, ce qui est correct dans ce cas.
- L'argument DefaultText n'est pas spécifié. Par conséquent, il s'agit d'une invite obligatoire.

```

select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#

```

### Invite obligatoire avec le type de données spécifié :

Tenez compte des points suivants :

- Cette invite nécessite une valeur entière valide comme réponse.
- L'argument DefaultText n'est pas spécifié. Par conséquent, il s'agit d'une invite obligatoire.

```

select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE >
  #prompt('Starting CountryCode',
    'integer',
    '',
    '',
    '')[gosales].[COUNTRY_MULTILINGUAL].[COUNTRY_CODE]')#

```

### Invite facultative et filtre obligatoire avec le type de données et la valeur par défaut spécifiés :

Tenez compte des points suivants :

- Cette invite permet à l'utilisateur de fournir une réponse sous la forme d'un entier valide.
- L'argument DefaultText est spécifié. Par conséquent, l'utilisateur peut omettre d'entrer une valeur, auquel cas la valeur 10 est utilisée. Cela se traduit par une invite facultative, mais pas par un filtre facultatif.

Exemple 1 :

```

select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE >
  #prompt('Starting CountryCode',
    'integer',
    '10'
  )#

```

Exemple 2 :

```

[gosales].[COUNTRY].[COUNTRY] = #prompt('countryPrompt','string','Canada')#
Result 2:
[gosales].[COUNTRY].[COUNTRY] = 'Canada'

```

Tenez compte des points suivants :

- Le paramètre `defaultText` doit être spécifié de manière à être littéralement valide dans le contexte de la macro, car aucun formatage n'est effectué sur cette valeur. Voir les détails ci-après.
- Dans l'exemple 2, la chaîne par défaut Canada est spécifiée sous la forme d'une chaîne utilisant une paire d'apostrophes, dans laquelle est intégrée une seconde paire d'apostrophes doublée, d'où la présence de 3 apostrophes. De cette manière, la chaîne s'affiche correctement dans l'expression, entre apostrophes simples.
- En règle générale, pour la chaîne `datatype`, le paramètre `defaultText` doit toujours être spécifié comme indiqué dans la remarque précédente, sauf dans le contexte d'un paramètre de procédure stockée.
- Pour le paramètre `defaultText` de type `date` ou `datetime`, un format spécial doit être utilisé dans le contexte SQL. Des exemples de ces formats sont `'DATE '2012-12-25''` et `'DATETIME '2012-12-25 12:00:00''`. Dans tous les autres contextes, utilisez `date/datetime` sans mot clé et entre apostrophes simples (par ex. : `'2012-12-25'`).

### Invite qui ajoute du texte à la valeur :

Tenez compte des points suivants :

- L'argument `DefaultText` est spécifié comme caractère espace. Dans ce cas, le texte généré est seulement le caractère espace qui élimine la clause `Where` de la requête.
- L'argument `Text` est spécifié et écrit dans le code SQL généré avant la valeur d'invite fournie par l'utilisateur.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#prompt('Starting CountryCode',
  'integer',
  ' ', // < = = this is a space
  'where COUNTRY_MULTILINGUAL.COUNTRY_CODE >'
)#
```

### Substitution de syntaxe :

Tenez compte des points suivants :

- L'argument `Datatype` est défini sur `token`, ce qui signifie que la valeur fournie par l'utilisateur est entrée sans aucun contrôle sur la valeur indiquée. Etant donné qu'aucune vérification n'est effectuée sur la valeur, l'éditeur d'expression peut indiquer que l'expression n'est pas valide. Lorsqu'une valeur fournie par l'utilisateur valide est indiquée ou si vous fournissez une valeur par défaut valide, l'éditeur d'expression interprète l'expression comme étant valide.
- `Token` doit être utilisé uniquement s'il existe une liste de valeurs de sélection pour l'utilisateur.
- L'argument `DefaultText` est spécifié. Par conséquent, il s'agit d'une invite facultative et `group by COUNTRY` est utilisé dans le code SQL généré.

```
Select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
```

```

from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#prompt('Sort column',
  'token',
  'group by COUNTRY',
  'group by '
)#

```

## Exemples : Création d'invites qui utilisent des tables de mappage des paramètres

Lorsqu'un rapport est exécuté, vous souhaitez que vos utilisateurs puissent sélectionner une langue pour les données de ce rapport. Les exemples ci-après illustrent plusieurs manières de procéder pour obtenir ce résultat.

### Invite qui utilise des variables de session :

Tenez compte des points suivants :

- Le nom de l'invite est spécifié à l'aide d'une correspondance de la table de mappage des paramètres nommée PromptLabels. La valeur de clé est la variable de session \$language.
- La clause Where est utilisée utilise une colonne paramétrée.

```

select
  ORDER_METHOD.ORDER_METHOD_CODE as ORDER_METHOD_CODE,
  ORDER_METHOD.ORDER_METHOD_#$language#
  as ORDER_METHOD_EN
from
  gosales.gosales.dbo.ORDER_METHOD ORDER_METHOD
#prompt($PromptLabels{$language},
  ''
  ,
  'where ORDER_METHOD.ORDER_METHOD_' + $language + '
>'
)#

```

### Table de mappage des paramètres avec des invites imbriquées :

Tenez compte des points suivants :

- Dans le modèle, il existe une table de mappage des paramètres DynPromptLabels avec # \$PromptLabels{\$language}#
- Certaines des informations d'invite sont exécutées à partir d'une table de mappage des paramètres au lieu d'être écrites directement dans le code SQL.
- La macro entière contenant l'invite peut être une valeur d'une table de mappage des paramètres.

```

select
  ORDER_METHOD.ORDER_METHOD_CODE as ORDER_METHOD_CODE,
  ORDER_METHOD.ORDER_METHOD_#$language#
  as ORDER_METHOD_EN
from
  gosales.gosales.dbo.ORDER_METHOD ORDER_METHOD
#prompt($DynPromptLabels{'ex9'},
  ''
  ,
  'where ORDER_METHOD.ORDER_METHOD_' + $language + '
>'
)#

```



## Exemples : Création d'une invite de choix à plusieurs valeurs

Lorsqu'un rapport est exécuté, vous souhaitez que vos utilisateurs puissent sélectionner une ou plusieurs valeurs. Les exemples ci-après illustrent plusieurs manières de procéder pour obtenir ce résultat.

### Invite avec un minimum requis :

Tenez compte des points suivants :

- L'utilisateur doit entrer au moins une valeur unique.
- Ceci ressemble au premier exemple d'invite pour un pays ou une région de la rubrique «Invite obligatoire sans type de données spécifié», à la page 171.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY IN (#promptmany('CountryName')#)
```

### Invite avec un minimum requis et le type de données spécifié :

Tenez compte des points suivants :

- Ceci ressemble au deuxième exemple d'invite pour un pays ou une région de la rubrique «Invite obligatoire avec le type de données spécifié», à la page 172.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (
  #promptmany('Selected CountryCodes',
    'integer',
    '',
    '',
    ''
  )#
  '[gosales].[COUNTRY_MULTILINGUAL].[COUNTRY_CODE]')#
)
```

### Invite facultative avec le type de données et la valeur par défaut spécifiés :

La clause In et les deux parenthèses font partie de l'instruction SQL.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN (
  #promptmany('Selected CountryCodes',
    'integer',
    '10'
  )#
)
```

### Invite qui ajoute du texte avant la syntaxe :

Tenez compte des points suivants :

- Cet exemple utilise l'argument TextFollowing.

```
select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
#promptmany('Selected CountryCodes',
  'integer',
  ' ', // < = = this
is a space
  'where COUNTRY_MULTILINGUAL.COUNTRY_CODE IN
( ',
  ' ',
  ')')
)#
```

### Invite facultative qui ajoute du texte avant la syntaxe :

```
Select
  COUNTRY_MULTILINGUAL.COUNTRY_CODE as COUNTRY_CODE,
  COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
  COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
  COUNTRY_MULTILINGUAL.CURRENCY_NAME as CURRENCY_NAME
from
  gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL,
  gosales.gosales.dbo.COUNTRY_XX
where COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE
#promptmany('Selected CountryCodes',
  'integer',
  ' ',
  ' and COUNTRY_MULTILINGUAL.COUNTRY_CODE IN ( ',
  ' ',
  ')')
)#
```

---

## Organisation des modèles

Lorsque vous organisez le modèle, vous permettez aux utilisateurs de rechercher et comprendre plus facilement les données contenues dans ce modèle. Vous pouvez également faciliter la gestion du modèle.

Une meilleure pratique consiste à créer plusieurs vues ou couches dans le modèle :

- Conservez les métadonnées de la source de données dans un espace-noms ou dossier distinct.

Dans IBM Cognos Framework Manager, il s'agit de la vue d'importation.

- Créez au moins un espace-noms ou dossier facultatif pour résoudre les difficultés relatives à l'interrogation à l'aide de sujets de requête ou d'objets dimensionnels.

Pour utiliser IBM Cognos Analysis Studio ou n'importe quelle requête OLAP, il doit y avoir un espace-noms ou un dossier dans le modèle qui représente les métadonnées avec objets dimensionnels.

- Créez au moins un espace-noms ou dossier pour la vue métier globale des métadonnées contenant des raccourcis vers les dimensions ou sujets de requête.

Dans Framework Manager, il s'agit de la vue métier. Utilisez des concepts métier pour modéliser la vue métier. Un même modèle peut contenir de nombreuses vues métier, chacune adaptée à un groupe d'utilisateurs différent. Publiez les vues métier.

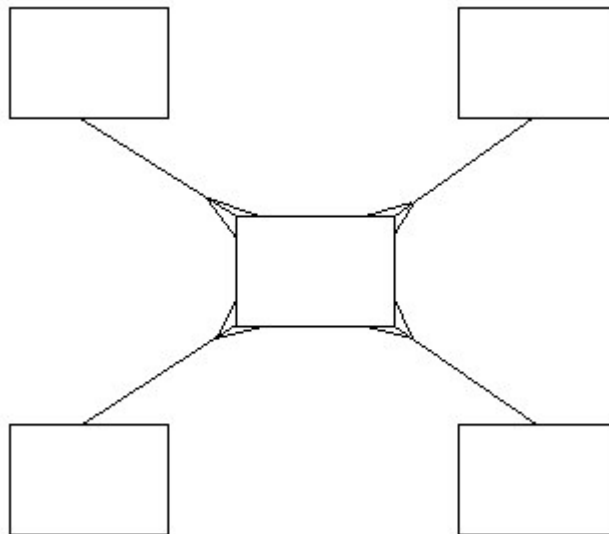
La sécurité peut être définie dans l'une des vues. Cela dépend de vos besoins métier. Par exemple, si devez empêcher tous les utilisateurs de visualiser un objet, vous ajoutez de la sécurité à ce dernier dans la vue d'importation. En règle générale, la sécurité est appliquée dans la vue métier.

## Création de groupes sous forme de schémas en étoile

Utilisez l'assistant **Créer un regroupement de schémas en étoile** pour créer rapidement des groupes de raccourcis qui vont fournir du contexte à vos utilisateurs sur les appartenances des objets entre eux. Cela rend le modèle plus intuitif aux yeux des utilisateurs. Les groupes sous forme de schémas en étoile peuvent également faciliter la génération de rapports portant sur des faits multiples en permettant la répétition des dimensions partagées dans différents groupes. Cela aide les utilisateurs à voir ce que les différents groupes ont en commun et la façon dont ils peuvent générer des rapports inter-fonctionnels ou à faits multiples. Pour plus d'informations, voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

Les groupes sous forme de schémas en étoile fournissent également du contexte pour les requêtes à plusieurs chemins de jointure. En créant des groupes sous forme de schémas en étoile dans la vue métier du modèle, vous pouvez déterminer quel chemin de jointure doit être sélectionné lorsqu'il y en a beaucoup. Cette fonctionnalité est particulièrement utile dans le cadre de requêtes sans fait.

Dans une conception de type schéma en étoile, les données transactionnelles numériques sont contenues dans une table de faits centrale à partir de laquelle les tables de dimension associées se propagent.



Les groupes sous forme de schémas en étoile peuvent contenir les dimensions, sujets de requête ou raccourcis sélectionnés. Les objets contenus dans un groupe sous forme de schéma en étoile ne peuvent pas référencer, ni être référencés par,

un objet se trouvant en dehors du groupe. Par conséquent, Framework Manager crée automatiquement un raccourci pour tout objet qui est référencé par un objet résidant en dehors du groupe.

Par exemple, dans un projet d'analyse de ventes, vous pouvez inclure les dimensions suivantes :

- dates des ventes (Heure)
- lieux des ventes (Région)
- détails produit (Produit)
- informations client (Client)

Vous ajoutez la quantité dans la table de faits.

## Création d'un schéma en étoile

### Procédure

1. Sélectionnez une ou plusieurs dimensions ou un ou plusieurs sujets de requête.
2. Cliquez sur **Outils, Créer un regroupement de schéma en étoile**.
3. Si vous souhaitez exclure un objet du groupe, dans la zone **Objets disponibles**, désélectionnez la case à cocher en regard de cet objet.
4. Effectuez l'une des opérations suivantes :
  - Pour ajouter des raccourcis au groupe, cliquez sur **Créer un raccourci pour tous les objets sélectionnés**.
  - Pour déplacer les objets vers le groupe, cliquez sur **Créer des raccourcis uniquement pour les objets utilisés hors du schéma en étoile**.
5. Pour déplacer les objets sélectionnés vers un espace-noms distinct, vérifiez que la case **Créer un espace-nom pour ce regroupement** est cochée et entrez le nom dans la zone **Nom de l'espace-noms**.
6. Cliquez sur **OK**.
7. S'il existe plusieurs relations, également appelés dimensions de rôles, créez des raccourcis de relation vers celles-ci («Création d'un raccourci de relation», à la page 74), ou créez des dimensions ou des sujets de requête individuels si vous devez les renommer.

### Groupe sous forme de schéma en étoile basé sur une dimension ou un sujet de requête unique

En général, vous devez sélectionner un seul objet lorsqu'il s'agit d'un fait qui dispose d'une relation à chaque dimension ou sujet de requête que vous souhaitez inclure dans le groupe sous forme de schéma en étoile. Lorsque vous créez un groupe sous forme de schéma en étoile qui est basé sur un objet unique, les événements suivants se produisent :

- Framework Manager affiche la liste des objets avec lesquels il dispose de relations pour que vous puissiez sélectionner rapidement les objets que vous souhaitez inclure dans le groupe.
- Le nom du groupe est basé sur le nom de la table de faits.
- Le nouveau groupe est créé sous le même parent que l'objet sélectionné.

### Groupe sous forme de schéma en étoile basé sur plusieurs dimensions ou sujets de requête

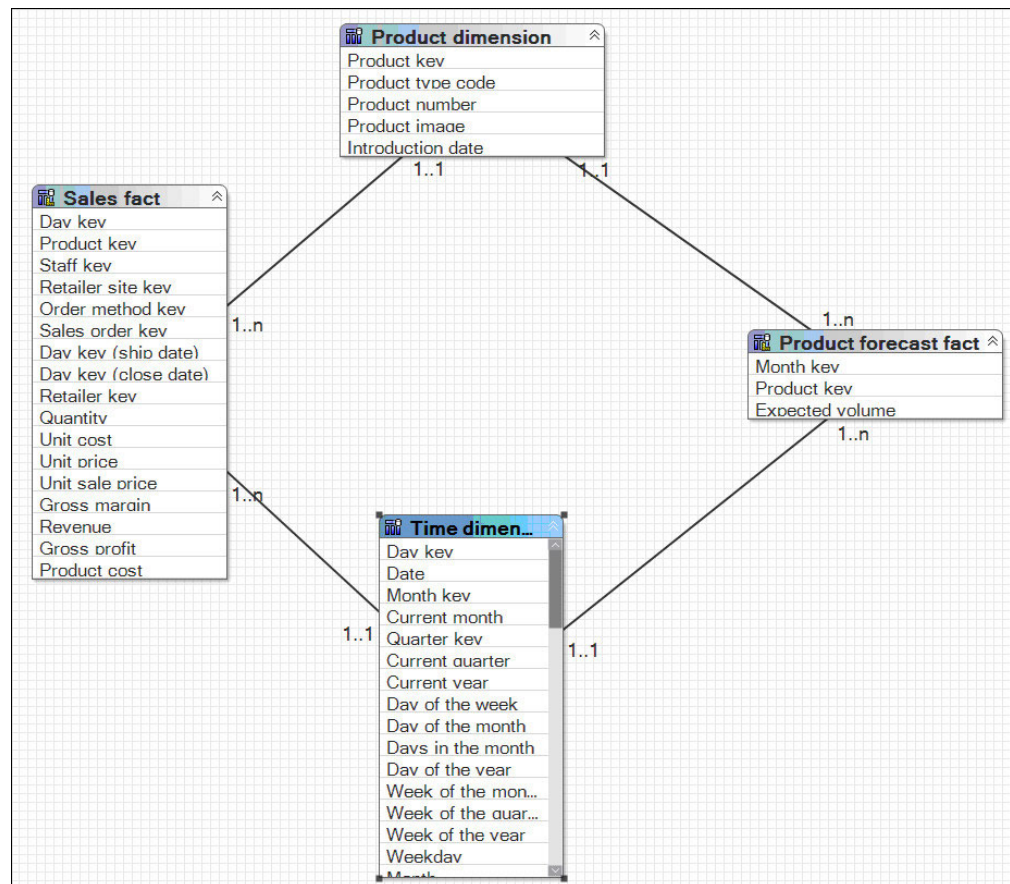
La sélection de plusieurs dimensions ou sujets de requête s'avère utile si vous souhaitez regrouper des dimensions ou des sujets de requête qui n'ont pas encore

de relations définies. Le nouveau groupe est placé sous le plus proche parent commun des dimensions ou des sujets de requête.

## Résolution de plusieurs schémas en étoile conformes

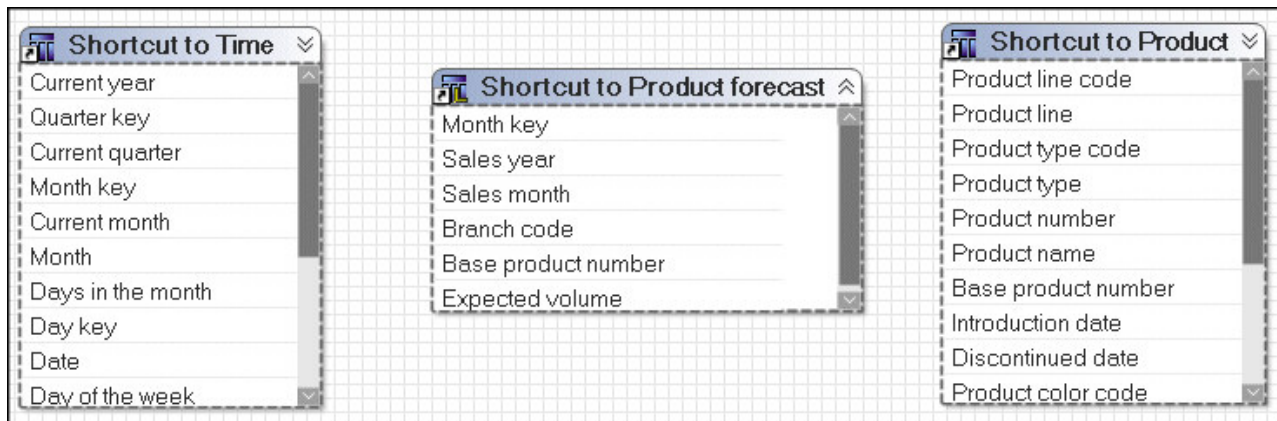
Vous verrez probablement des sujets de requête dimensionnels qui sont joints à plusieurs sujets de requête de faits. L'ambiguïté au niveau des jointures est un problème pouvant se produire lors de la création de rapports à partir d'éléments issus de plusieurs dimensions ou sujets de requête dimensionnels sans inclure aucun élément de la dimension de mesure ou du sujet de requête de fait. On parle alors de requête sans fait.

Par exemple, les dimensions Produit et Temps sont associées aux faits Prévisions de vente et Ventes.



A l'aide de ces relations, comment créer un rapport utilisant uniquement des éléments de Produit et Année ? La question professionnelle pourrait être de savoir quels produits ont été prévus à la vente en 2005 ou quels produits ont effectivement été vendus en 2005. Même si cette requête implique uniquement Produit et Temps, ces dimensions sont associées par le biais de plusieurs faits. Il n'existe aucun moyen de savoir quelle question professionnelle est posée. Vous devez définir le contexte pour la requête sans fait.

Dans l'exemple suivant, il est conseillé de créer deux espaces-noms, un contenant des raccourcis vers Produit, Temps et Prévisions de vente, l'autre contenant des raccourcis vers Produit, Temps et Ventes.



Lorsque vous procédez ainsi pour tous les schémas en étoile, vous résolvez les ambiguïtés de jointure en plaçant les raccourcis vers le fait et toutes les dimensions dans un même espace-noms. Les raccourcis associés aux dimensions conformes de chaque espace-noms sont identiques et constituent des références vers l'objet d'origine. **Remarque** : Il s'agit exactement de la même règle que celle appliquée aux dimensions ordinaires et aux dimensions de mesure.

Avec un espace-noms pour chaque schéma en étoile, vos utilisateurs n'ont plus aucun doute concernant les éléments à utiliser. Pour créer un rapport sur les produits réellement vendus en 2005, ils utilisent les éléments **Produit** et **Année** de l'espace-noms **Ventes**. La seule relation pertinente dans ce contexte est celle associant les éléments **Produit**, **Temps** et **Ventes** ; elle est utilisée pour renvoyer les données.

#### Résolution d'un schéma en étoile :

##### Procédure

1. Sélectionnez l'une des dimensions de mesure.
2. Dans le menu **Outils**, cliquez sur **Créer un regroupement de schéma en étoile**.
3. Si vous souhaitez exclure un objet du groupe, dans la zone **Objets disponibles**, désélectionnez la case à cocher en regard de cet objet.
4. Cliquez sur **Créer un raccourci pour tous les objets sélectionnés**.

5. Pour déplacer les objets sélectionnés vers un espace-noms distinct, vérifiez que la case **Créer un espace-nom pour ce regroupement** est cochée et entrez le nom dans la zone **Nom de l'espace-noms**.
6. Cliquez sur **OK**.
7. Répétez ces étapes pour l'autre dimension de mesure.

## **Modélisation de dimensions en flocons en tant que dimensions en étoile**

Une dimension en flocons supprime les attributs de cardinalité de bas niveau des tables de dimension et les place dans des dimensions secondaires qui sont liées aux dimensions d'origine par des clés artificielles.

Pour modéliser une dimension en flocons en tant que dimension en étoile, procédez comme suit:

### **Procédure**

1. Sélectionnez les sujets de requête requis pour accéder aux données et placez-les dans un nouvel espace-noms («Création de dossiers ou d'espaces-noms», à la page 185).
2. Vérifiez que toutes les relations sont correctes («Relations», à la page 68).
3. Traitez les métadonnées multilingues («Métadonnées multilingues», à la page 125).
4. Créez une dimension de modèle pour chaque dimension en flocons :
  - Sélectionnez les sujets de requête dont vous avez besoin.
  - Cliquez sur **Actions, Fusionner dans la nouvelle dimension ordinaire**.
  - Renommez la nouvelle dimension de modèle. Par défaut, son nom est composé des noms concaténés des objets d'origine.
5. Si vous avez besoin de plusieurs niveaux, effectuez l'une des opérations suivantes :
  - Utilisez la grille des dimensions pour définir des hiérarchies et des niveaux pour la dimension («Création de dimensions ordinaires», à la page 108).
  - Indiquez des déterminants si les niveaux sont stockés dans un seul sujet de requête («Déterminants», à la page 84).

## **Raccourcis**

Un raccourci est un pointeur vers un objet, par exemple, un rapport, une dimension, un sujet de requête ou un dossier. Utilisez des raccourcis dans la vue métier lorsque des groupes d'utilisateurs se chevauchent et que vous souhaitez inclure les métadonnées dans plusieurs dossiers. Avec des raccourcis, vous pouvez disposer de plusieurs références à un objet.

Par exemple, vous créez des dossiers nommés Commandes, Produits et Clients. Si vous souhaitez que les dossiers Commandes et Clients contiennent la même dimension, vous devez créer un raccourci vers la dimension et l'ajouter aux deux dossiers.

**Remarque :** Deux raccourcis vers des espaces-noms ou des dossiers ne doivent pas avoir le même nom dans un modèle. Pour les autres types de raccourci (par exemple, un raccourci de sujet de requête), le nom doit être unique au sein de l'espace-noms parent.

Lorsque vous créez un raccourci, IBM Cognos Framework Manager ne définit pas les propriétés **Infobulle** et **Description**. Si vous ne définissez pas ces propriétés, les

valeurs affichées dans les studios IBM Cognos sont les mêmes que celles définies dans l'objet auquel le raccourci fait référence.

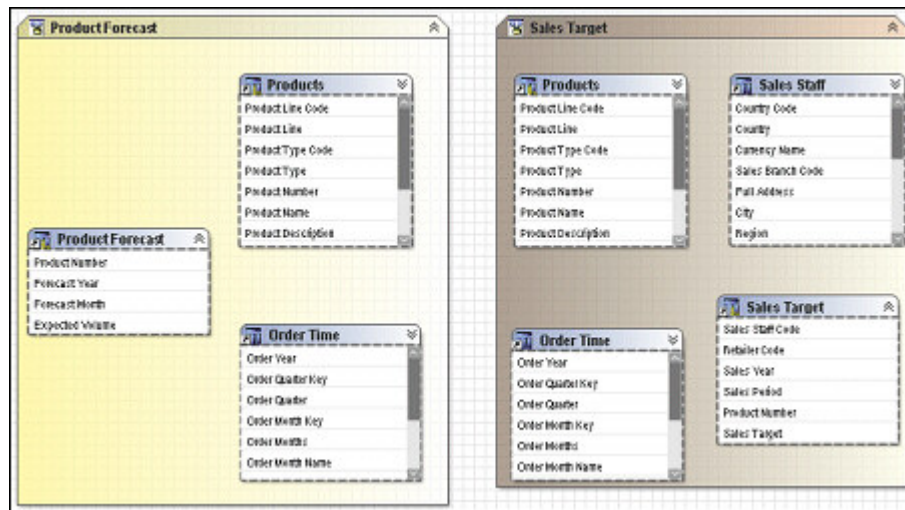
**Astuce :** Pour accéder à l'objet auquel le raccourci fait référence, cliquez avec le bouton droit de la souris sur le raccourci et cliquez sur **Accéder à la cible**.

Les raccourcis sont moins souples que les objets de modèle du point de vue de la présentation, mais leur maintenance est beaucoup plus légère car ils sont automatiquement mis à jour dès que l'objet cible est mis à jour. Si la maintenance est l'une de vos principales préoccupations et que vous n'avez pas besoin de personnaliser l'apparence du sujet de requête, utilisez des raccourcis.

IBM Cognos Framework Manager dispose de deux types de raccourci :

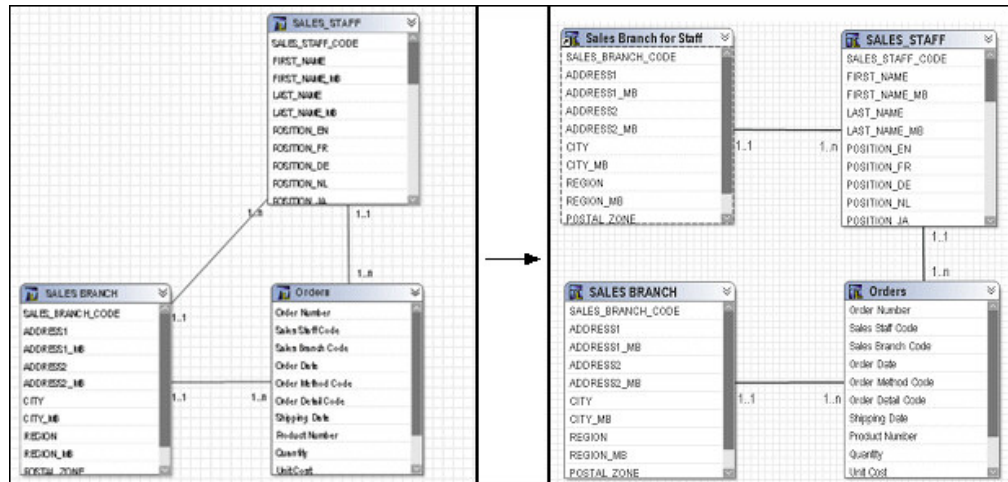
- les raccourcis ordinaires, qui sont une simple référence à l'objet cible ;
- les raccourcis d'alias, qui se comportent comme s'ils étaient une copie de l'objet d'origine avec un comportement complètement indépendant. Les raccourcis d'alias ne sont disponibles que pour les sujets de requête et les dimensions.

Les raccourcis ordinaires sont généralement utilisés en tant que dimensions conformes avec groupes sous forme de schémas en étoile, en créant plusieurs références avec exactement les mêmes nom et apparence à plusieurs endroits. Dans l'exemple ci-dessous, les raccourcis créés pour Produits et Heure de commande se comportent comme des références. Si une requête est écrite pour obtenir Produits à partir de Prévisions de vente et Objectifs de vente, la requête utilise la définition de Produits basée sur le raccourci d'origine et cette définition figure une seule fois dans la requête.



Les raccourcis d'alias sont généralement utilisés dans les dimensions de rôle ou les tables partagées. Il y a déjà un exemple relatif aux dimensions de rôle dans le présent document, nous allons donc étudier le cas des tables partagées. Dans cet exemple, Personnel de vente et Succursale peuvent être considérés comme des hiérarchies distinctes. Notre connaissance des données nous permet de savoir que les membres du personnel peuvent changer de succursale, il faut donc que nous puissions produire des rapports sur Succursale et Personnel de vente à la fois indépendamment et de façon groupée. Pour ce faire, nous devons créer un alias de Succursale pouvant être utilisé en tant que niveau dans la hiérarchie Personnel de vente.





Une fois le nouveau raccourci d'alias en place, il est possible de créer des requêtes qui demandent les commandes par succursale et les commandes par équipe commerciale, avec en même temps les informations relatives à leur succursale actuelle.

## Raccourcis et relations

Lorsque vous décidez à quel endroit placer des raccourcis, réfléchissez à la manière dont la portée des raccourcis affecte les relations. Les raccourcis placés dans un dossier autre que celui du sujet de requête cible utilisent les relations du sujet de requête cible. Les raccourcis placés dans le même dossier que le sujet de requête cible ignorent les relations du sujet de requête cible et utilisent uniquement celles indiquées pour le raccourci.

Vous pouvez spécifier pour un raccourci une relation autre que celles de l'objet de requête cible. En créant des relations du raccourci vers d'autres sujets de requête, vous évitez les erreurs de jointure en chassé-croisé dans le modèle.

Vous ne pouvez pas créer de raccourcis vers des relations de portée.

## Raccourcis et dimensions ou sujets de requête

Les raccourcis se traduisent par moins de dimensions ou de sujets de requête à gérer. Vous pouvez conserver des dimensions ou des sujets de requête dans la vue d'importation et conserver des raccourcis dans la vue métier. Les raccourcis sont le plus fréquemment utilisés lors de la création de groupes sous forme de schémas en étoile.

Lorsque vous créez un raccourci vers une dimension ou un sujet de requête, vous ne pouvez pas personnaliser les éléments de requête qui se trouvent dans le raccourci. La totalité de la dimension ou du sujet de requête est incluse dans le raccourci.

Lorsque vous ouvrez un modèle d'une édition précédente, la restriction **Traitement des raccourcis** a pour valeur **Automatique**. Lorsque l'option **Automatique** est utilisée, les raccourcis fonctionnent de la même façon que dans les éditions précédentes, c'est-à-dire qu'un raccourci qui se trouve dans le même répertoire que sa cible se comporte comme un alias, ou une instance indépendante, alors qu'un raccourci existant ailleurs dans le modèle se comporte comme une référence au

raccourci d'origine. Pour tirer parti de la propriété **Traiter en tant que**, il est conseillé de vérifier le modèle, et au moment de la réparation, d'affecter la valeur **Jointure explicite** à la restriction. La réparation modifie tous les raccourcis en leur affectant la valeur correcte à partir de la propriété **Traiter en tant que** selon les règles suivies par le paramètre **Automatique**. Cela signifie que le comportement de votre modèle ne changera pas tant que vous ne modifierez pas la propriété **Traiter en tant que** de vos raccourcis.

Lorsque vous créez un nouveau modèle, la valeur de la restriction **Traitement des raccourcis** est toujours **Jointure explicite**.

Lorsque la restriction a pour valeur **Jointure explicite**, le comportement du raccourci est dicté par la propriété **Traiter en tant que** et vous avez un contrôle total sur la façon dont les raccourcis se comportent sans avoir à connaître leur emplacement dans le modèle.

Des raccourcis peuvent être créés à l'aide de l'assistant **Créer un regroupement de schémas en étoile**. Par exemple, une table de faits et ses tables de dimension sont stockées dans la vue d'importation. Si vous souhaitez représenter des dimensions conformes dans plusieurs groupes sous forme de schémas en étoile, il ne peut exister qu'une seule dimension ou qu'un seul sujet de requête pour chaque table de dimension. Utilisez des raccourcis pour toutes les autres instances de la dimension ou du sujet de requête. En utilisant des raccourcis, vous pouvez créer des requêtes impliquant plusieurs tables de faits qui sont associées par le biais de tables de dimension partagées.

Dans le cas de dimensions de rôle, il existe deux approches principales décrites dans la rubrique «Dimensions de rôles», à la page 350. Vous pouvez créer un sujet de requête et des dimensions ordinaires pour chaque rôle, puis utiliser des raccourcis en tant que références dans des groupes sous forme de schémas en étoile pour permettre l'utilisation des dimensions de rôle en tant que dimensions conformes, ou créer un raccourci pour chaque rôle dans chaque groupe sous forme de schéma en étoile et créer une jointure entre chaque raccourci de rôle et le raccourci de fait associé. Tant que la cible des raccourcis ne se trouve pas dans le même dossier, tous les raccourcis se comportent comme des alias.

La sécurité que vous spécifiez pour un objet est transmise aux raccourcis qui font référence à l'objet sécurisé. Si vous disposez d'un raccourci vers un objet sécurisé, seuls les utilisateurs autorisés à afficher l'objet sécurisé peuvent voir le raccourci dans le pack publié.

Si un raccourci est défini vers une dimension ou un sujet de requête, vous pouvez spécifier le comportement du raccourci dans la propriété **Traiter en tant que**. Vous pouvez définir la propriété sur l'une des valeurs suivantes :

- Référence  
Utilisez cette option lorsque vous souhaitez qu'une réplique exacte d'un sujet de requête dans plusieurs emplacements se comporte comme un objet unique si elle est référencée dans la même requête.
- Alias  
Utilisez cette option lorsque vous souhaitez qu'une réplique exacte d'un sujet de requête se comporte comme un objet indépendant suivant un chemin de jointure indépendant. Les chemins de jointure indépendants doivent d'abord être définis dans le modèle.

La restriction **Traitement des raccourcis** contrôle le comportement de tous les raccourcis. La restriction **Traitement des raccourcis** a la priorité sur la propriété **Traiter en tant que**. Par exemple, si la restriction est définie sur **Automatique**, le comportement du raccourci est déterminé par l'emplacement du raccourci par rapport à sa cible, quelle que soit la valeur de la propriété **Traiter en tant que**. Pour plus d'informations, voir «Restrictions», à la page 309.

## Procédure de création de raccourcis

### Procédure

1. Cliquez avec le bouton droit de la souris sur les sujets de requête, les dimensions ou les dossiers vers lesquels vous souhaitez créer des raccourcis et effectuez l'une des opérations suivantes :
  - Cliquez sur **Créer, Raccourci d'alias**.
  - Cliquez sur **Créer, Raccourci**.
  - Cliquez sur **Créer un regroupement de schémas en étoile**. Cette commande est également disponible dans le menu **Outils**.
2. Pour des raccourcis vers des sujets de requête ou des dimensions, dans la sous-fenêtre **Propriétés**, définissez la propriété **Traiter en tant que** sur **Alias** ou **Référence**.

## Création de dossiers ou d'espaces-noms

Vous pouvez créer des dossiers ou des espaces-noms afin d'organiser les objets dans le modèle.

L'information la plus importante à connaître à propos des espaces-noms est qu'une fois la création du rapport commencée, tout changement de nom d'un espace-noms publié aura une incidence sur votre contenu IBM Cognos. En effet, le changement du nom de l'espace-noms modifie les identificateurs des objets publiés dans les métadonnées. Etant donné que l'espace-noms fait partie de l'identificateur d'objet dans IBM Cognos Framework Manager, chaque espace-noms doit avoir un nom unique dans le modèle. Chaque objet de l'espace-noms doit également avoir un nom unique. Une partie de la stratégie d'élaboration des groupes sous forme de schémas en étoile est de placer les raccourcis dans un espace-noms distinct, ce qui crée automatiquement un identificateur unique pour chaque objet dans l'espace-noms. Pour les bases de données relationnelles, cela vous permet d'utiliser le même nom pour les raccourcis vers des dimensions conformes dans différents groupes sous forme de schémas en étoile.

Lors de la prochaine tentative d'exécution d'une requête, d'un rapport ou d'une analyse sur le modèle mis à jour, vous obtenez une erreur. Si vous devez renommer l'espace-noms que vous avez publié, utilisez la commande **Analyze Publish Impact** pour identifier les rapports sur lesquels cela aura une incidence.

Les dossiers sont beaucoup moins compliqués que les espaces-noms. Ils n'ont qu'une fonction purement organisationnelle et n'ont aucune incidence sur l'identificateur des objets ou sur votre contenu. Vous pouvez créer des dossiers pour organiser les objets par sujet ou domaine fonctionnel. Cela facilite l'accès aux métadonnées, en particulier dans les projets volumineux.

Le principal inconvénient des dossiers réside dans le fait qu'ils nécessitent des noms uniques pour tous les sujets de requête, dimensions et raccourcis. Ils ne représentent donc pas la solution idéale pour contenir des objets partagés.

**Astuce :** Lorsque vous affichez des métadonnées sous l'onglet **Diagramme**, vous pouvez développer ou réduire les dossiers et les espaces-noms. A partir du menu **Diagramme**, cliquez sur **Tout réduire** ou **Tout développer**.

Si vous définissez la sécurité sur un dossier dans lequel vous placez ensuite des objets, vérifiez que les exclusions ont été correctement définies.

## Procédure de création d'un dossier

### Procédure

1. Cliquez sur **Actions, Créer, Dossier**.
2. Dans la zone **Nom du dossier**, attribuez un nom au nouveau dossier.
3. Cliquez sur **Suivant**.
4. Indiquez si vous souhaitez déplacer les objets ou pour créer des raccourcis :
  - Pour déplacer les objets sélectionnés dans le dossier, cliquez sur **Déplacer les éléments sélectionnés vers le nouveau dossier**. Lorsque vous déplacez un objet qui participe à une relation, la relation à cet objet est également déplacée.
  - Pour créer des raccourcis faisant référence aux objets sélectionnés, cliquez sur **Créer un raccourci pour les objets sélectionnés**. Ne sélectionnez pas tous les objets de l'espace-noms pour éviter de créer une structure récursive dans le pack publié.
5. Sélectionner les données à ajouter au dossier.
6. Cliquez sur **Terminer**.

## Procédure de création d'un espace-noms

### Procédure

1. Cliquez sur **Actions, Créer, Espace-noms**.
2. Cliquez avec le bouton droit de la souris sur l'espace-noms, cliquez sur **Renommer** et attribuez à l'espace-noms un nom descriptif unique.
3. Ajoutez des objets en important des métadonnées, ou en déplaçant vers l'espace-noms des objets de modèle ou des raccourcis vers des objets.

## Création de dossiers d'élément de requête

Si vous créez un dossier d'élément de requête, vous ne pouvez pas changer l'ordre des éléments de requête dans la boîte de dialogue **Editer la définition** pour le sujet de requête de modèle. Vous pouvez changer l'ordre des éléments uniquement dans l'**afficheur de projet**.

Vous pouvez créer des dossiers d'élément de requête pour organiser des sujets de requête ou des dimensions contenant un grand nombre d'éléments de requête. Un dossier d'élément de requête peut contenir uniquement des éléments de requête et des dossiers d'élément de requête.

## Procédure de création d'un dossier d'éléments de requête

### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, cliquez sur un sujet de requête ou une dimension.
2. Cliquez sur **Actions, Créer, Dossier d'éléments de requête**.

Un nouveau dossier d'éléments de requête s'affiche dans **Afficheur de projet** sous les éléments de requête qui appartiennent à ce sujet de requête ou cette dimension.

3. Faites glisser les éléments de requête souhaités vers le dossier d'élément de requête.

Vous ne pouvez pas ajouter des éléments de requête qui n'existent pas dans le sujet de requête ou la dimension parent.

## Création de dossiers de mesures

Vous pouvez créer des dossiers de mesures pour organiser des dimensions de mesure qui contiennent de nombreux éléments de requête. Vous pouvez imbriquer des dossiers de mesures dans d'autres dossiers de mesures.

Vous ne pouvez pas créer de dossier de mesures à partir d'un raccourci de mesure.

Si vous créez un dossier de mesures, vous ne pouvez pas changer l'ordre des mesures dans la boîte de dialogue **Editer la définition** pour la dimension de mesure. Vous pouvez changer l'ordre des mesures uniquement dans l'**afficheur de projet**.

### Procédure de création d'un dossier de mesures

#### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, cliquez sur une dimension de mesure.
2. Cliquez sur **Actions, Créer, Dossier des mesures**.

Un nouveau dossier s'affiche dans **Afficheur de projet** sous les mesures qui appartiennent à cette dimension de mesure.

3. Faites glisser les éléments de requête souhaités vers le dossier de mesures.

Vous ne pouvez pas ajouter des mesures qui n'existent pas dans la dimension de mesure parent.

---

## Modèles durables

Lorsque vous construisez un modèle, vous devez envisager la possibilité que les besoins de l'utilisateur changent avec le temps et que vous soyez amené à répercuter ces modifications dans le modèle. Votre objectif est de créer un modèle flexible capable de résister aux modifications nécessaires sans impacter les rapports existants, les auteurs de rapports, et les utilisateurs finaux.

L'une des modifications les plus fréquentes que les modélisateurs doivent implémenter dans leurs modèles est le changement de nom des éléments de requête. Si vos modèles sont durables, ces types de modification peuvent être apportés rapidement sans impacter les rapports existants. Les calculs et les filtres qui font référence aux éléments de requête renommés restent valides.

Les modèles durables sont utiles dans les environnements multilingue et unilingue lorsque le changement de nom des éléments de requête fait suite à une modification des besoins métier. Dans un environnement multilingue, les modèles durables simplifient également le processus de traduction en vous permettant de spécifier des libellés spécifiques de la langue pour des éléments de requête sans risquer de rompre les références de rapport qui existent pour d'autres langues.

Lorsque vous utilisez des modèles durables, n'oubliez pas les consignes suivantes :

- Spécifiez une langue de conception pour votre projet.

Choisissez la version des paramètres régionaux de la langue qui n'est pas incluse dans vos besoins métier. Il peut s'agir de paramètres régionaux moins fréquemment utilisés, tel que Anglais (Zimbabwe). Par exemple, lorsque vous créez un projet avec la langue de conception Anglais (Zimbabwe), la langue active est également Anglais (Zimbabwe). Vous ne pouvez pas modifier la langue de conception d'un projet une fois que vous avez créé un modèle.

- Lorsque vous concevez le modèle, la langue de conception et la langue active doivent être les mêmes.

La conception inclut la création de nouveaux éléments tels que les espaces-noms et les sujets de requête, l'inclusion de calculs ou la modification de la structure du modèle. Par exemple, pour les activités de conception, la langue de conception et la langue active devront toutes deux être Anglais (Zimbabwe)

- Lorsque vous renommez les éléments de modèle pour vos utilisateurs métier, la langue active ne doit pas être la même que la langue de conception.

Lorsque vous êtes prêt à modifier les noms des éléments de requête, remplacez les paramètres de langue active par une langue autre que la langue de conception. Par exemple, lorsque vous renommez les éléments de requête, votre langue de conception est Anglais (Zimbabwe). Votre langue active pourra être Allemand.

- Si votre langue active est différente de la langue de conception, ne modifiez pas la structure de votre modèle.

Pour effectuer des modifications structurelles, vous devez faire en sorte que la langue active soit la même que la langue de conception avant d'effectuer les modifications.

Pour rendre votre modèle durable, attribuez la valeur True à la propriété de projet **Utiliser l'environnement local de conception pour l'identificateur de référence**.

**Remarque :** les traitements de modèles durables ne sont pas pris en charge pour la modélisation dimensionnelle des sources de données relationnelles.

Pour plus d'informations sur les modèles durables, voir la section Conseils éprouvés du centre de support d'IBM Cognos ([http://www.ibm.com/software/data/support/cognos\\_crc.html](http://www.ibm.com/software/data/support/cognos_crc.html)).

## Création d'un modèle durable

### Procédure

1. Dans la page **Bienvenue**, cliquez sur **Créer un nouveau projet**.  
**Astuce :** Si vous êtes dans IBM Cognos Framework Manager, cliquez sur **Nouveau** dans le menu **Fichier**.
2. Dans la page **Nouveau projet**, spécifiez un nom et un emplacement pour le projet et cliquez sur **OK**.
3. Dans la page **Sélectionner la langue**, cliquez sur la langue de conception du projet, puis sur **OK**.  
Veillez à sélectionner la langue de conception appropriée, comme indiqué précédemment dans cette section. Vous ne pouvez pas modifier la langue après avoir cliqué sur **OK**, mais vous pouvez ajouter ultérieurement d'autres langues de projet.
4. Dans l'**assistant de métadonnées**, cliquez sur **Suivant** pour importer vos métadonnées.
5. Suivez les instructions de l'**assistant de métadonnées** :

- Sélectionnez une connexion de source de données et cliquez sur **Suivant**.  
Si la connexion de source de données souhaitée n'est pas répertoriée, vous devez d'abord la créer («Création de connexions de sources de données», à la page 43).
- Cochez les cases correspondant aux objets que vous souhaitez importer.
- Spécifiez la manière dont l'importation doit traiter les noms d'objet en double. Indiquez si vous souhaitez importer et créer un nom unique. Si vous choisissez de créer un nom unique, l'objet importé s'affiche avec un numéro. Par exemple, vous verrez SujetRequête et SujetRequête1 dans votre projet.
- Si vous souhaitez importer des objets système, cochez la case **Afficher les objets système**, puis sélectionnez les objets système à importer.
- Spécifiez les critères à utiliser pour créer des relations et cliquez sur **Importer**.

Pour plus d'informations, voir «Relations», à la page 68.

Une liste des objets qui n'ont pas pu être importés et le nombre d'objets importés s'affichent.

6. Cliquez sur **Terminer**.

Enregistrez le fichier projet (.cpf) et tous les fichiers associés dans un seul dossier. Si vous enregistrez un projet sous un autre nom ou un autre format, veillez à enregistrer le projet dans un dossier distinct.

7. Cliquez sur le nom du projet dans l'**afficheur de projet** et définissez la propriété de projet **Utiliser l'environnement local de conception pour l'identificateur de référence** sur **true**.

**Remarque:** Si vous réattribuez ultérieurement la valeur **false** à cette propriété après avoir renommé des éléments de requête dans le modèle, les rapports basés sur ce modèle seront altérés.

8. Cliquez sur **Projet, Langues, Définir les langues**, puis ajoutez les langues requises au projet en choisissant l'une d'entre elle comme **langue active**.

Lorsque vous spécifiez la langue active, veillez à ce qu'elle ne soit pas la même que la langue de conception. Pour plus d'informations, voir «Ajout d'une langue à un projet», à la page 129.

9. Enregistrez le projet.

10. Créez les packs requis et publiez-les dans IBM Cognos Connection.

Utilisez les packs publiés pour créer un contenu dans IBM Cognos Report Studio, IBM Cognos Query Studio ou IBM Cognos Event Studio. Par exemple, créez des rapports dans Report Studio ou Query Studio.

## Résultats

Vous pouvez maintenant tester le modèle pour vérifier qu'il fonctionne comme prévu.

## Test d'un modèle durable

### Procédure

1. Lancez Report Studio en utilisant le pack publié dans Création d'un modèle durable, puis créez et enregistrez un rapport.
2. Dans Framework Manager, ouvrez le projet créé dans Création d'un modèle durable et vérifiez que la langue active du projet est différente de la langue de conception.

3. Dans la langue active, renommez certains des éléments de requête inclus dans le pack publié à l'étape 1.
4. Republiez le pack dans IBM Cognos Connection en remplaçant le pack d'origine.
5. Lancez Report Studio en utilisant le pack republié et ouvrez le rapport créé à l'étape 1.

Le rapport affiche les noms d'élément de requête modifiés dans la sous-fenêtre **Objets insérables**. Cependant, la spécification de rapport affiche les noms d'élément de requête dans la langue de conception, pas dans la langue active, dans laquelle vous avez effectué les modifications.

6. Exécutez le rapport.

Les colonnes représentant les éléments de requête renommés affiche les nouveaux noms.

## Résultats

Le modèle est durable car le fait de renommer des éléments de requête n'altère pas des rapports existants.

---

## Analyse de modèles

Vous pouvez analyser les métadonnées contenues dans un modèle à l'aide de **Model Advisor**, outil automatisé qui applique des règles basées sur les instructions de modélisation en cours et identifie les zones du modèle que vous devez examiner. Pour vous aider à comprendre la nature du problème mis en évidence, ainsi que certaines des actions possibles, des liens vous sont fournis vers les sections appropriées de la documentation. **Model Advisor** n'est pas un outil de remplacement pour un modélisateur averti ; il fournit aux nouveaux modélisateurs un outil d'assistance et offre aux modélisateurs expérimentés un outil de diagnostic.

Vous pouvez sélectionner un ou plusieurs tests à exécuter sur le modèle sélectionné ou sur un sous-ensemble d'un modèle. Vérifiez le modèle et corrigez les erreurs avant de l'analyser. Pour plus d'informations, voir «Vérification des modèles ou des packs», à la page 255.

Si vous analysez un modèle nouveau, utilisez le flux de travaux suivant :

- Analysez les objets qui viennent d'être importés, notamment leurs relations et déterminants.
- Utilisez les anomalies identifiées pour résoudre les problèmes potentiels de génération de requête.
- Lorsque vous générez des vues supplémentaires, utilisez **Model Advisor** pour analyser chacune d'elles en vue de détecter des problèmes potentiels.
- Avant de publier le modèle, utilisez **Model Advisor** sur les objets qui seront publiés.

Si vous analysez un modèle établi plus ancien ou un modèle qui n'est pas encore terminé, utilisez **Model Advisor** pour valider les pratiques de modélisation. Le flux de travaux est semblable à celui utilisé pour les nouveaux modèles : commencer par la vue de base de données, puis poursuivre.



## Faits identifiés par la cardinalité

Ce test recherche les sujets de requête qui possèdent uniquement la cardinalité multiple (n) sur les extrémités de la relation dans laquelle ils sont impliqués. Les sujets de requête ayant cette cardinalité sont traités comme des faits lors de la génération des requêtes, par conséquent, il est essentiel de faire en sorte qu'ils soient correctement identifiés.

## Sujets de requête qui peuvent se comporter comme des faits ou des dimensions

Ce test recherche les sujets de requête qui possèdent une cardinalité mixte : multiple (n) et un ou zéro (1, 0). La cardinalité mixte renvoie au comportement d'un sujet de requête qui est susceptible de changer en fonction des autres sujets de requête utilisés dans une requête. Cela peut générer des requêtes imprévisibles dans certains cas. Si le sujet de requête est évalué en tant que fait, il est inclus dans le chemin de la requête. S'il est évalué en tant que dimension, il peut être ignoré s'il n'est pas directement référencé dans la requête. Pour garantir un comportement prévisible, vous devez résoudre ces scénarios.

La cardinalité mixte peut désigner un sujet de requête qui fait parti d'une dimension en flocons ou d'une relation Principale/Détails. En présence de tables intermédiaires dans les dimensions en flocons, cela ne pose aucun problème à résoudre. Des problèmes risquent de survenir dans les situations où plusieurs chemins de requête sont disponibles ; le test doit mettre en évidence ces situations pour les sujets de requête ayant des relations multiples.

## Sujets de requête avec plusieurs relations

Ce test recherche les sujets de requête qui comportent de nombreuses relations entre deux objets ou une jointure en boucle qui ne représente pas le motif de jointure de schéma en étoile.

Les relations multiples entre deux sujets de requête sont souvent associées à des dimensions de rôles. Avec les dimensions de rôles, vous créez des alias qui permettent à chaque rôle de disposer d'une jointure différente et d'avoir un comportement indépendant différent. Les chemins de jointure multiples qui désignent des jointures de boucle (sauf pour les schémas en étoile) peuvent provoquer des problèmes relatifs à des requêtes fractionnées de façon incorrecte. Cela a une incidence sur la prévisibilité de la génération de requêtes pour vos utilisateurs.

Si le logiciel IBM Cognos comporte des relations multiples sans critères de distinction permettant de choisir l'une d'entre elles, il utilise la relation qui se classe en premier dans l'ordre alphabétique. Si vous devez créer une requête qui utilise une relation différente, un problème subsiste. De même, si vous souhaitez utiliser des filtres sur les critères propres au rôle défini par les relations et que ces filtres s'excluent mutuellement, aucune donnée n'est affichée dans le rapport. Pour plus d'informations sur les dimensions de rôles et pour obtenir un exemple, voir «Dimensions de rôles», à la page 350.

**Remarque :** Si la cardinalité identifie clairement les faits et dimensions, le logiciel IBM Cognos peut automatiquement résoudre les jointures de boucles causées par des données de schémas en étoile lorsque plusieurs tables de faits sont jointes à un ensemble commun de tables de dimensions. **Model Advisor** ignore les motifs de

jointure de schéma en étoile car ils ne sont pas considérés comme des jointures problématiques.

## Sujets de requête joints à eux-mêmes

Ce test recherche des relations réflexives et récursives. Ces relations impliquent un minimum de deux niveaux de granularité. IBM Cognos Framework Manager importe les relations réflexives mais ne les utilise pas lors de l'exécution de requêtes. Les relations réflexives, également appelées jointures automatiques, sont affichées dans le modèle à titre représentatif uniquement.

La plupart du temps, un sujet de requête joint à lui-même désigne une relation parent-enfant. Les fichiers qui ont des relations parent-enfant peuvent avoir une profondeur définie ou indéfinie. La seule façon de savoir à quoi vous en tenir consiste à comprendre les données et le concept métier représentés.

Une relation parent-enfant peut être source de problème ou non, selon la manière dont vous souhaitez utiliser les données dans votre application. Dans certains cas, il se peut qu'un seul niveau de la relation soit nécessaire, et cela implique un scénario plutôt simple à modéliser. Si vous connaissez les données qui doivent remplir complètement chaque niveau et qui doivent avoir un nombre précis de niveaux, vous pouvez également choisir une solution de modélisation uniquement. Toutefois, pour les cas où le volume de données est important, les données ne sont pas complètement remplies à chaque niveau (hiérarchie décalée ou non équilibrée) et/ou le nombre de niveaux dans les données peut changer au fil du temps, il est recommandé de transformer les données dans une structure à plat avec un nombre défini de colonnes avant de procéder à la modélisation dans Framework Manager.

## Déterminants en conflit avec des relations

Ce test recherche les déterminants qui entrent en conflit avec la relation définie pour le sujet de requête. Les déterminants permettent de garantir que l'agrégation dans les rapports est correcte et que les requêtes sont générées correctement.

Les déterminants reflètent la granularité en représentant des sous-ensembles ou des groupes de données dans un sujet de requête et ils permettent de garantir l'agrégation correcte de ces données répétées. Les déterminants sont étroitement liés au concept de clés et d'index de la source de données et sont importés en fonction des informations de clé et d'index uniques contenues dans cette source de données. Vérifiez toujours les déterminants importés et, si nécessaire, modifiez-les ou créez-en d'autres. Lorsque vous modifiez les déterminants, vous pouvez redéfinir les informations de clé et d'index contenues dans votre source de données, en les remplaçant par des informations qui correspondent mieux à vos besoins en matière d'analyse et de génération de rapports. En ajoutant des déterminants, vous pouvez représenter des groupes de données répétées qui sont pertinentes pour votre application.

**Model Advisor** vérifie les sujets de requête comportant des déterminants et il signale ceux qui répondent à l'un des critères suivants :

- La relation fait référence à tous les éléments de requête d'un déterminant unique et la cardinalité de la relation n'est pas 1:1 ni 0:1.
- La relation fait référence à certains des éléments de requête d'un déterminant unique et la cardinalité de la relation n'est pas 1:n ni 0:n.

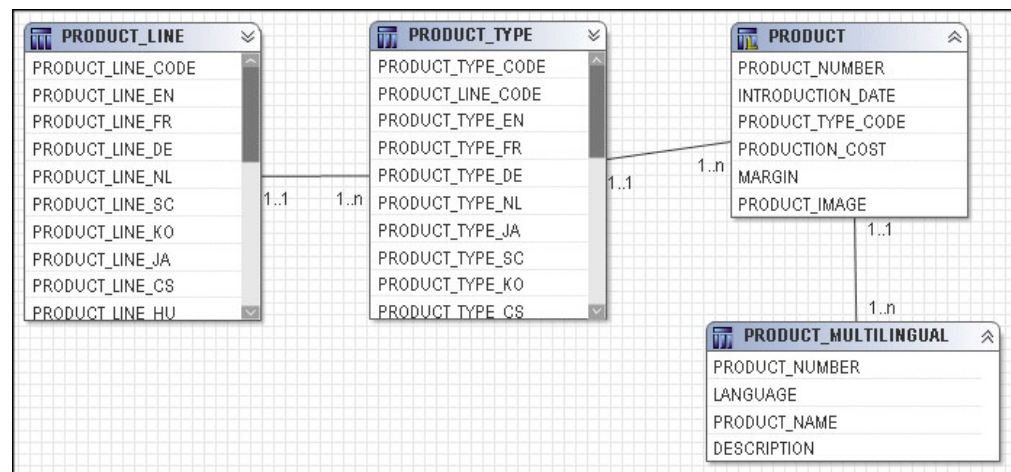
Model Advisor signale également les occurrences dans lesquelles les clés d'une relation ne correspondent pas aux clés d'un groupe par déterminant.

## Facteurs qui remplacent le paramètre de langage SQL réduit

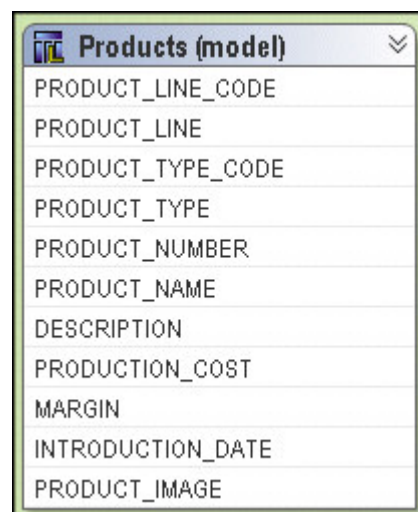
Ce test recherche les divers facteurs qui remplacent la valeur Réduit pour le paramètre de type de génération SQL, tels que les sujets de requête de source de données modifiés, les relations entre les sujets de requête de modèle ou les déterminants de sujets de requête de modèle.

Lorsque vous utilisez le langage SQL réduit, le code SQL généré contient uniquement l'ensemble de tables et de jointures minimum nécessaire à l'obtention des valeurs pour les éléments de requête sélectionnés.

Voici un exemple illustrant la signification de SQL réduit. Cet exemple présente quatre sujets de requête, Ligne de produits, Type de produit, Produit et Produit multilingue, qui sont joints les uns aux autres.



Ils peuvent être combinés dans un sujet de requête de modèle.



Par exemple, si vous testez le sujet de requête Produits (modèle) dans son ensemble, vous pouvez voir que la clause from de la requête fait référence à quatre tables.

```

select
  PRODUCT_LINE.PRODUCT_LINE_CODE as Product_Line_Code,
  PRODUCT_LINE.PRODUCT_LINE_EN as Product_Line,
  PRODUCT_TYPE.PRODUCT_TYPE_CODE as Product_Type_Code,
  PRODUCT_TYPE.PRODUCT_TYPE_EN as Product_Type,
  PRODUCT.PRODUCT_NUMBER as Product_Number,
  PRODUCT_MULTILINGUAL.PRODUCT_NAME as Product_Name
  PRODUCT_MULTILINGUAL.DESCRPTION as Product_Description,
  PRODUCT.INTRODUCTION_DATE as Introduction_Date,
  PRODUCT.PRODUCT_IMAGE as Product_Image,
  PRODUCT.PRODUCTION_COST as Production_Cost,
  PRODUCT.MARGIN as Margin
from
  gos1_82..gos1.PRODUCT_LINE PRODUCT_LINE,
  gos1_82..gos1.PRODUCT_TYPE PRODUCT_TYPE,
  gos1_82..gos1.PRODUCT PRODUCT,
  gos1_82..gos1.PRODUCT_MULTILINGUAL PRODUCT_MULTILINGUAL
where
  (PRODUCT_MULTILINGUAL."LANGUAGE" - N'EN')
  and
  (PRODUCT_LINE.PRODUCT_LINE_CODE = PRODUCT_TYPE.PRODUCT_LINE_CODE)
  and
  (PRODUCT_TYPE.PRODUCT_TYPE_CODE = PRODUCT.PRODUCT_TYPE_CODE)
  and
  (PRODUCT.PRODUCT_NUMBER = PRODUCT_MULTILINGUAL.PRODUCT_NUMBER

```

Si vous testez uniquement Nom de produit, vous pouvez constater que la requête qui en résulte utilise seulement Produit multilingue, qui est la table requise. Il s'agit des effets du SQL réduit.

```

select
  PRODUCT_MULTILINGUAL.PRODUCT_NAME as Product_Name
from
  gos1_82..gos1.PRODUCT_MULTILINGUAL PRODUCT_MULTILINGUAL
where
  (PRODUCT_MULTILINGUAL."LANGUAGE" - N'EN')

```

## Calculs intégrés qui utilisent le type d'agrégation calculée

Ce test détecte les situations où vous avez défini la propriété **Agrégat ordinaire** avec la valeur **Calculé** pour les calculs imbriqués.

Le type d'agrégation calculée est pris en charge uniquement pour les éléments suivants :

- les calculs autonomes ;
- les calculs qui sont imbriqués dans des dimensions de mesure et basés sur des mesures provenant de la même dimension de mesure.

Pour plus d'informations sur les agrégations calculées, voir «Règles d'interprétation des agrégations calculées», à la page 138.

## Sujets de requête pouvant provoquer un conflit de mise en cache de métadonnées

Ce test recherche les facteurs qui remplacent les métadonnées mises en cache, tels que les sujets de requête de source de données dont le code SQL a été modifié ou les sujets de requêtes qui contiennent des calculs ou des filtres.

IBM Cognos Framework Manager stocke les métadonnées importées de la source de données. Cependant, selon les paramètres de restriction et certaines actions que vous effectuez dans le modèle, ces métadonnées peuvent ne pas être utilisées lors

de la préparation d'une requête. Si vous sélectionnez la restriction **Autoriser la portabilité du modèle amélioré lors de l'exécution**, Framework Manager interroge systématiquement la source de données pour obtenir des informations sur les métadonnées avant de préparer une requête. Si vous n'avez pas sélectionné pas la restriction **Autoriser la portabilité du modèle amélioré lors de l'exécution**, Framework Manager accède aux métadonnées qui ont été stockées dans le modèle au lieu d'interroger la source de données. Les principales exceptions sont les suivantes :

- Toute modification du code SQL dans un sujet de requête de source de données. Cela inclut l'utilisation de macros.
- L'ajout d'un calcul ou d'un filtre à un sujet de requête de source de données.

**Remarque :** les requêtes de métadonnées générées par IBM Cognos sont bien prises en charge par la plupart des fournisseurs de systèmes de gestion de bases de données relationnelles et ne devraient donc pas avoir une incidence notable sur la majeure partie des applications de génération de rapports.

**Model Advisor** est uniquement destiné à être utilisé avec des modèles de métadonnées relationnelles. N'exécutez pas **Model Advisor** sur la totalité d'un modèle, mais appliquez-le individuellement à certaines vues pour faire en sorte que le retour d'informations soit bien pris en compte. Par exemple, si un problème dans une vue d'importation n'a pas été traité, il peut être résolu par l'opération de modélisation qui est effectuée dans une vue intermédiaire.

Pour les modèles ou les espaces-noms volumineux, il se peut que **Model Advisor** ne renvoie pas de résultats immédiatement.

Etant donné que **Model Advisor** n'est pas sensible aux données, vous devez connaître les données, le modèle et les métadonnées appropriés pour vos besoins en termes d'information décisionnelle. Les éléments signalés par **Model Advisor** ne sont pas tous problématiques. Le contexte de chaque problème soulevé par **Model Advisor** est important.

## Procédure d'analyse d'un modèle

### Procédure

1. Cliquez sur un ou plusieurs objets à analyser.
  - Sélectionnez des sujets de requête, des dimensions, des hiérarchies, des calculs, des éléments de requête ou des raccourcis pour analyser les objets qui s'afficheront dans un rapport spécifique afin de tester le rapport avant sa création.
  - Sélectionnez un dossier ou un espace-noms pour analyser tous ses objets. Si un objet fait référence à un objet d'un autre dossier ou espace-noms, l'objet référencé est également analysé.
  - Sélectionnez un pack avant de le publier pour vous assurer qu'il respecte les bonnes pratiques de modélisation.
2. Cliquez sur **Outils, Exécuter Model Advisor**.

**Astuce :** Vous pouvez également cliquer avec le bouton droit de la souris sur un ou plusieurs objets, puis cliquer sur **Exécuter Model Advisor**.
3. Dans l'onglet **Options**, sélectionnez les critères à utiliser pour l'analyse.
4. Cliquez sur **Analyser**.
5. Dans l'onglet **Model Advisor**, examinez les problèmes qui ont été identifiés.

Chaque problème est décrit, un lien vers des informations supplémentaires sur chaque problème est disponible et tous les objets impactés par chacun de ces problèmes sont répertoriés.

6. Pour savoir si un objet connaît un problème, cliquez sur l'icône de **Context**

**Explorer**  dans la colonne **Action** du rapport.

L'explorateur de contexte **Context Explorer** affiche les objets auxquels l'objet sélectionné est connecté. Vous pouvez sélectionner un objet associé et voir à quels objets il est connecté.

---

## Chapitre 6. Métadonnées SAP BW

Remarque : Les informations concernant les métadonnées relationnelles figurent dans une autre rubrique.

Après avoir importé des métadonnées, vous devez vous assurer qu'elles sont configurées selon les besoins de vos utilisateurs en termes de rapports et fournir les informations supplémentaires requises. Les améliorations que vous apportez à IBM Cognos Framework Manager n'ont pas d'incidence sur la source de données d'origine.

**Astuce :** Pour vérifier que le modèle répond aux besoins en termes de rapports, vous pouvez sélectionner des objets à afficher dans un rapport et les tester. Les résultats du test montrent le rapport que vos utilisateurs verront, ainsi que le code SQL et les messages issus du logiciel IBM Cognos, le cas échéant. Vous pouvez également publier un pack à tout moment, puis l'utiliser pour créer des rapports.

Vous pouvez vérifier le projet quand bon vous semble pour vous assurer que les références entre les objets qu'il contient sont valides. Voir «Vérification des modèles ou des packs», à la page 255.

Vous pouvez effectuer les tâches suivantes lorsque vous travaillez avec des métadonnées SAP BW dans IBM Cognos Framework Manager :

- Importer les métadonnées.
- Utiliser des dimensions.
- Contrôler la manière dont les données sont utilisées et formatées en cochant des propriétés d'élément de requête.
- Si nécessaire, ajouter des règles métier supplémentaires, par exemple, des calculs et des filtres, pour affiner les données extraites et mettre les informations appropriées à disposition des utilisateurs.
- Organiser le modèle en créant pour chaque groupe d'utilisateurs des vues distinctes reflétant les concepts métier que les utilisateurs connaissent bien.
- Si nécessaire, ajuster des paramètres de Framework Manager et de studios IBM Cognos pour optimiser les performances.

Après avoir travaillé sur le modèle, vous pouvez créer un pack et le publier pour vos utilisateurs. Pour plus d'informations, voir «Publication de packs», à la page 269.

**Remarque :** Vous pouvez également créer des packs pour des cubes et des requêtes SAP BW directement dans IBM Cognos Connection. Pour plus d'informations, voir la section sur les packs du *Guide d'administration et de sécurité*.

---

### Importation à partir d'une source de données SAP BW

Lorsque vous effectuez une importation à partir d'une source de données SAP BW, vous pouvez importer toutes les métadonnées ou importer uniquement les objets que vous sélectionnez.

Pour plus d'informations concernant le mappage d'objets métadonnées SAP BW sur des objets IBM Cognos Framework Manager, voir «Mappage d'objets SAP BW sur des objets Framework Manager», à la page 203.

Vous pouvez souhaiter disposer de différentes vues (ou couches) dans le modèle : une vue d'importation destinée à contenir les métadonnées que vous avez importées à partir de la source de données et une vue métier dans laquelle vous améliorez les métadonnées. Après l'importation, vous pouvez copier les métadonnées dans la vue métier. Vous disposez alors de deux vues qui doivent être synchronisées avec BW InfoProvider.

**Astuce :** si vous souhaitez exposer les chiffres clés calculés à partir d'une requête SAP BW, vous devez importer cette dernière.

## Accès aux métadonnées et données SAP BW

Lors de l'utilisation d'une source de données SAP BW, le fait que les utilisateurs accèdent à des métadonnées InfoCube ou InfoQuery n'implique pas qu'ils puissent également accéder aux données contenues dans ces objets. Pour permettre à Framework Manager d'extraire des métadonnées de SAP BW, vous devez configurer des privilèges d'accès dans le système SAP BW. Pour faire en sorte que les utilisateurs disposent des droits d'accès appropriés, vérifiez les autorisations affectées aux rôles des utilisateurs.

Les objets d'autorisation suivants doivent être configurés de manière à permettre à Framework Manager d'importer des cubes d'informations ou des sources de données, appelés InfoCubes dans le système SAP BW.

- S\_RFC
    - Définissez la zone **Activité** sur la valeur : 16
    - Définissez la zone **Nom de RFC à protéger** sur la valeur : SYST, RSOB, SUGU, RFC1, RS\_UNIFICATION, RSAB, SDTX, SU\_USER, RSNDI\_SHIE
    - Définissez l'objet **Type de RFC** en vue de désigner la zone protégée sur la valeur : FUGR
  - S\_TABU\_DIS
    - Définissez la zone **Activité** sur la valeur : 03
    - Définissez le **Groupe d'autorisation** sur la valeur : &NC&
- Remarque :** &NC& représente une table n'ayant pas de groupe d'autorisation. Pour plus de sécurité, créez un groupe d'autorisation et affectez-lui la table RSHIEDIR. Le nouveau groupe d'autorisation limite l'accès de l'utilisateur à la table ci-dessus uniquement qui est nécessaire à l'outil de modélisation. Créez le groupe d'autorisation en tant que personnalisation dans le système SAP.
- S\_USER\_GRP
    - Définissez la zone **Activité** sur la valeur : 03, 05
    - Définissez la zone **Groupe d'utilisateurs pour maintenance principale des utilisateurs** sur la valeur par défaut.
  - S\_RS\_COMP
    - Définissez la zone **Activité** sur la valeur par défaut.
    - Dans la zone **InfoArea**, entrez *Nom technique InfoArea*
    - Dans la zone **InfoCube**, entrez *Nom technique InfoCube*
    - Définissez la zone **Nom (identificateur) des composants de génération de rapports** sur la valeur par défaut.



Définissez la zone **Type de composants de génération de rapports** sur la valeur par défaut.

- S\_RS\_COMP1

Définissez la zone **Activité** sur la valeur par défaut.

Définissez la zone **Nom (identificateur) des composants de génération de rapports** sur la valeur par défaut.

Définissez la zone **Type de composants de génération de rapports** sur la valeur par défaut.

Définissez la zone **Propriétaire (personne responsable)** sur la valeur par défaut.

- S\_RS\_HIER

Dans la zone **Activité**, entrez 71.

Dans la zone **Nom de hiérarchie**, entrez la valeur *Nom de hiérarchie*

Dans la zone **InfoObject**, entrez *Nom technique InfoObject*.

Dans la zone **Version**, entrez *Version de hiérarchie*.

- S\_RS\_ICUBE

Définissez la zone **Activité** sur la valeur : 03

Définissez la zone **sous-objet InfoCube** sur les valeurs : DATA et DEFINITION

Dans la zone **InfoArea**, entrez *Nom technique InfoArea*

Dans la zone **InfoCube**, entrez *Nom technique InfoCube*.

### Astuces

- &NC& représente toute table non dotée d'un groupe d'autorisation. Pour des raisons de sécurité, créez un nouveau groupe d'autorisation et affectez-lui la table RSHIEDIR. Le nouveau groupe d'autorisation limite l'accès de l'utilisateur à la table ci-dessus uniquement, qui est requise par Framework Manager. Créez le nouveau groupe d'autorisation en tant que personnalisation dans le système SAP BW.
- Vous pouvez utiliser l'astérisque (\*) pour représenter toutes les valeurs, lorsqu'il est affiché de façon autonome, ou des valeurs partielles, lorsqu'il est utilisé n'importe où dans une chaîne.

## Structures SAP BW

De nombreuses requêtes SAP BW existantes contiennent des structures que vous pouvez utiliser dans des requêtes IBM Cognos pour contrôler la quantité et l'ordre des informations disponibles pour vos utilisateurs. Par exemple, avec des structures doubles, vous pouvez créer un rapport de type tableau croisé avec une structure sur chaque axe.

Les structures sont les suivantes :

- Structure de chiffre clé

SAP BW Query Designer crée automatiquement une structure de chiffre clé lorsque vous ajoutez des chiffres clés à une requête. Vous devez disposer d'au moins un chiffre clé pour importer les métadonnées de requête dans IBM Cognos Framework Manager, même si vous n'utilisez pas de chiffre clé dans les rapports. Par conséquent, vous devez toujours disposer d'une structure de chiffre clé.

- Structure de caractéristique

Une structure de caractéristique est une collection de valeurs de caractéristique (membres) issues d'une ou plusieurs dimensions. Vous pouvez créer une structure de caractéristique dans SAP en ajoutant une structure à la requête, puis

en ajoutant les membres requis à la structure. Dans le logiciel IBM Cognos, la structure s'affiche sous la forme d'une dimension supplémentaire ne comportant qu'un niveau multiracine.

Lorsque vous importez la requête SAP BW dans Framework Manager, la structure de chiffre clé apparaît dans la dimension de mesure appelée Chiffres clés et la structure de caractéristique s'affiche sous la forme d'une dimension supplémentaire.

Si vous réimportez la même requête SAP BW dans Framework Manager, vous devez utiliser le même paramètre pour la case à cocher **Prise en charge des structures doubles SAP BW**. Framework Manager ne vous permet pas de sélectionner un paramètre différent pour la même requête, car des objets différents seraient alors générés dans le modèle, ce qui se traduirait par des erreurs. Vous pouvez utiliser des paramètres différents pour des requêtes différentes.

## Hiéarchies SAP BW

Lorsque vous importez des métadonnées, Framework Manager génère une dimension dans chaque caractéristique SAP BW.

Une seule hiérarchie associée à une caractéristique donnée peut être utilisée dans un rapport. Par conséquent, vous devez grouper les dimensions qui représentent les hiérarchies d'une caractéristique unique dans un dossier ou un sujet de requête de modèle pour faciliter la génération de rapports pour vos utilisateurs.

S'il existe plusieurs hiérarchies dans une source de données SAP BW, la première hiérarchie qui est importée devient la hiérarchie par défaut.

Framework Manager prend en charge les types de hiérarchie suivants :

- Caractéristique  
Il s'agit d'une de toutes les valeurs de caractéristique.
- Noeud de texte  
Les noeuds non-feuilles contiennent uniquement du texte et ne font référence à aucun autre objet de source de données.
- Valeur de caractéristique  
Les noeuds de chaque niveau d'une hiérarchie de présentation sont des valeurs issues d'une autre caractéristique.
- Récursive  
Les noeuds de la totalité de la hiérarchie de présentation proviennent de la caractéristique proprement dite.

Si une caractéristique n'est pas dans une dimension de temps, mais qu'il s'agit d'une date et qu'elle est traitée en tant que date dans SAP BW, la caractéristique est importée avec le type de données date.

Framework Manager ne prend pas en charge les hiérarchies qui contiennent plusieurs types de noeud. Ces hiérarchies sont importées mais elles sont masquées dans le modèle Framework Manager.

Comme les métadonnées hiérarchiques sont automatiquement générées pour SAP BW, vous ne pouvez pas les modifier dans Framework Manager.

## Hiéramchies versionnées

Vous pouvez importer les types de hiéramchies versionnées suivants à partir d'une source de données SAP BW :

- Hiéramchie dépendante de la version

Une hiéramchie peut comprendre plusieurs versions. Chaque version d'une hiéramchie peut avoir une structure différente, par exemple, Ventes par région et Ventes par responsable. Lors de l'importation de métadonnées, Framework Manager identifie chaque version en tant que hiéramchie distincte et crée une dimension pour chacune d'entre elles.

- La totalité de hiéramchie est associée à une contrainte horaire

Chaque version est associé à une plage de temps sans chevauchement avec une autre version de la même hiéramchie. La structure de chaque version peut être différente. Lors de l'importation des métadonnées, Framework Manager identifie chaque version en tant que hiéramchie et inclut la plage de temps dans le nom de dimension.

- Structure de hiéramchie avec contrainte horaire

Il existe une version unique de la hiéramchie, mais des plages de temps applicables peuvent être affectées à des noeuds au sein de la hiéramchie. La structure de la hiéramchie peut changer au fur et à mesure que de nouveaux niveaux sont insérés ou retirés. Par exemple, des niveaux qui représentent différents districts de vente peuvent être ajoutés au fil du temps. Lors de l'importation de métadonnées, Framework Manager identifie une structure hiéramchique avec contrainte horaire en tant que hiéramchie non versionnée et reconnaît la structure de la hiéramchie à la date en cours.

Le type de dimension détermine la hiéramchie utilisée et, pour les hiéramchies avec contrainte horaire, la date à utiliser pour contrôler la version.

Framework Manager définit la date de clé de requête de hiéramchies avec contrainte horaire en fonction des dates qui font partie de la hiéramchie avec contrainte horaire. Vous pouvez alors sélectionner des versions spécifiques de hiéramchies. Pour les hiéramchies avec des versions avec contrainte horaire, la valeur par défaut est la date et l'heure en cours. La hiéramchie que vous appliquez à une caractéristique dépend du type de date de clé de requête : fixe, en cours ou variable. La date de clé de requête est définie pour une date spécifique.

Pour une date fixe, incluez uniquement la version qui correspond à la date fixe dans la requête SAP BW sous-jacente. Par exemple, si la requête BW SAP a une date fixe telle que 2005, seule la version correspondant à 2005 est importée.

Pour une date en cours, incluez uniquement la version qui correspond à l'intervalle de temps correspondant au présent jusqu'à un futur raisonnablement proche.

Pour une variable, définissez la date pour la variable dans Framework Manager et incluez uniquement la version de la hiéramchie applicable à cette date.

Lorsque vous utilisez Framework Manager pour modéliser des données SAP BW, les versions ou les dates appliquées à une hiéramchie de présentation dans SAP BW ne sont pas importées dans le modèle. Par conséquent, toutes les versions de la hiéramchie sont accessibles dans Framework Manager.

Vous pouvez disposer d'une hiéramchie avec contrainte horaire et d'une variable définie dans SAP BW pour établir la date d'entrée en vigueur pour la hiéramchie.

Dans ce cas, affectez une date fixe à la variable dans Framework Manager et incluez uniquement la dimension correspondant à cette date dans le modèle.

Dans Framework Manager, si une hiérarchie versionnée n'est pas associée à une contrainte horaire et comprend une version fixe, incluez uniquement la version de la hiérarchie associée à la version sélectionnée. Sinon, la hiérarchie présentée à vos utilisateurs sera inaccessible.

## Procédure d'accès à un InfoCube sécurisé

### Procédure

1. Créez dans Business Explorer Query Designer une requête qui accède à l'InfoCube.
2. Créez une variable d'autorisation pour chaque InfoObjet de l'InfoCube sous-jacent pour lequel il existe des autorisations.
3. Pour chaque variable, vérifiez que l'option Ready for Input est désactivée. Par défaut, cette option est activée.
4. Activez la requête pour un accès via OLE DB pour OLAP.
5. Enregistrez la requête.
6. Dans Framework Manager, faites référence à la requête et non à l'InfoCube.

## Procédure d'importation à partir d'une source de données SAP BW

### Procédure

1. Vérifiez qu'une connexion est établie avec la source de données.  
Pour plus d'informations sur la création de connexions de source de données, voir le *Guide d'administration et de sécurité IBM Cognos*.
2. Cliquez sur l'espace-noms, le dossier ou le segment vers lequel vous souhaitez effectuer l'importation, puis dans le menu **Actions**, cliquez sur **Exécuter l'assistant de métadonnées**.
3. Sélectionnez **Sources de données** et cliquez sur **Suivant**.
4. Sélectionnez une connexion de source de données SAP BW et cliquez sur **Suivant**.
5. Sélectionnez les objets à importer.  
Si vous effectuez une réimportation, l'objet existant est mis à jour.  
Si vous importez un nouvel objet et qu'il existe un objet portant le même nom, le nouvel objet est importé et un numéro est ajouté au nom d'origine. Par exemple, vous verrez SujetRequête et SujetRequête1 dans votre projet.  
Une fois que des éléments de requête sont importés, vous ne pouvez pas les supprimer sans supprimer la totalité du sujet de requête.
6. Si vous souhaitez importer une structure de caractéristique et une structure de chiffre clé, cochez la case **Prise en charge des structures doubles SAP BW**.  
Le contenu de la page **Sélectionner les objets** est mis à jour pour refléter les structures doubles de votre source de données.  
**Remarque :** Si vous réimportez la même requête SAP BW (par exemple, parce que la source de données sous-jacente a été modifiée), vous devez utiliser la même valeur pour cette case à cocher. Vous ne pouvez pas utiliser un paramètre différent pour la même requête, car des objets différents seraient alors générés dans le modèle, ce qui se traduirait par des erreurs.

7. Sélectionnez les langues à importer.  
Les langues doivent exister dans la source de données.  
Vous pourrez ajouter ultérieurement des langues à votre projet, mais vous ne pourrez pas revenir en arrière et importer les métadonnées spécifiques de la langue à partir de la source de données. Une fois l'importation terminée, vous devez ajouter manuellement les métadonnées spécifiques de la langue.
8. Indiquez si vous souhaitez que Framework Manager affiche le nom abrégé, le nom long ou le nom technique pour les dimensions.
  - Si vous sélectionnez le nom abrégé et que la zone est vide, le nom long est affiché.
  - Si vous sélectionnez le nom long et que la zone est vide, le nom abrégé est affiché.
  - Si vous sélectionnez le nom abrégé ou le nom long et que les deux zones sont vides, le nom technique est affiché.
9. Pour organiser les objets du modèle de la même manière que dans Business Explorer Query Designer, sélectionnez l'option **Améliorer le pack pour l'organisation SAP BW des objets**.  
Vous aurez alors un dossier pour chaque caractéristique.
10. Cliquez sur **Suivant**.  
Une liste des objets qui n'ont pas pu être importés s'affiche avec le nombre d'objets importés.
11. Cliquez sur **Terminer**.

## Résultats

Après l'importation, vérifiez les valeurs de la propriété d'utilisation et d'agrégation. Les table de faits peuvent contenir des colonnes numériques qui ne doivent pas être agrégées, par exemple, des taux de change.

Si vous souhaitez recréer une requête sur un autre système SAP BW, utilisez le mécanisme de migration SAP BW pour le transport de la requête. Cela permet de garantir que le nom technique de chaque mesure reste le même. Tout projet faisant référence à la requête sera ainsi dirigé vers l'un des systèmes sans modification du projet.

---

## Mappage d'objets SAP BW sur des objets Framework Manager

Les objets SAP BW sont mappés sur les objets IBM Cognos Framework Manager suivants.

Objet SAP BW	Objet Framework Manager
Requête, InfoCube, RemoteCube, MultiCube	Espace-noms.

Objet SAP BW	Objet Framework Manager
Caractéristique	<p>Dossier contenant des dimensions.</p> <p>Vous devez sélectionner l'option <b>Améliorer le modèle pour l'organisation des objets dans SAP BW</b> lors de l'importation de métadonnées afin de disposer d'un dossier pour chaque caractéristique.</p> <p><b>Remarque :</b> par défaut, Framework Manager importe les caractéristiques Devise et Unité de mesure SAP BW. Vous pouvez supprimer ces caractéristiques si elles ne vous sont d'aucune utilité.</p>
Dimension	<p>Dimension. La dimension peut contenir des hiérarchies représentant chaque hiérarchie de présentation. La hiérarchie par défaut contient deux niveaux représentant</p> <ul style="list-style-type: none"> <li>- l'agrégation de toutes les valeurs de caractéristique, également appelée valeur Tous ;</li> <li>- toutes les valeurs de caractéristique.</li> </ul>
Chiffre clé	Elément de requête qui fait partie d'une dimension de mesure appelée Chiffres clés.
Niveau de hiérarchie de présentation	<p>Niveau.</p> <p><b>Remarque :</b> les noms de niveau doivent être définis dans le plan de travail d'administration pour être significatifs.</p>
Attribut	Elément de requête associé à un niveau dont la valeur de la propriété <b>Syntaxe est Attribut</b> .
Variable SAP BW	<p>Propriété de source de données.</p> <p>Pour plus d'informations sur les variables SAP BW prises en charge par Framework Manager, voir «Variables SAP BW», à la page 233.</p>

Pour plus d'informations sur la définition des privilèges d'accès en vue d'extraire des métadonnées de SAP BW, voir «Accès aux métadonnées et données SAP BW», à la page 198.

---

## Dimensions (SAP BW)

Une dimension représente un grand groupe de données sur un aspect essentiel d'une entreprise, tel que des produits, des dates ou des marchés.

Les types de dimensions que vous pouvez gérer dans IBM Cognos Framework Manager sont les dimensions ordinaires et les dimensions de mesure. Dans SAP BW, les dimensions de mesure sont appelées chiffres clés.

Par exemple, dans un projet d'analyse des ventes, vous incluez ces dimensions :

Nom	Saisir	Description
Heure	Dimension ordinaire	Dates de ventes organisées en les années, les trimestres, les mois, les semaines et les jours où les ventes ont été effectuées.
Région	Dimension ordinaire	Emplacements des ventes groupés en régions, pays et villes de vente
Produit	Dimension ordinaire	Détails sur les produits organisés par type de produit, marque, modèle, couleur et emballage.
Client	Dimension ordinaire	Informations client
Ventes	Dimension de chiffre clé	Détails des achats tels que les unités vendues, les revenus et le bénéfice

## Modification d'une dimension ordinaire (SAP BW)

Une dimension ordinaire contient des informations de clé métier et des informations descriptives classées selon une hiérarchie, du niveau de granularité le plus élevé au plus faible. Elle dispose généralement de plusieurs niveaux qui doivent tous avoir une clé et une légende. Si vous n'avez pas de clé unique pour votre niveau, il est conseillé d'en créer une dans un calcul.

Les dimensions ordinaires de modèle sont basées sur des sources de données ou des sujets de requête de modèle déjà définis dans le modèle. Vous devez définir une clé métier et une légende de type de chaîne pour chaque niveau. Lorsque vous vérifiez le modèle, toute absence de clé métier et de légende est détectée. Au lieu de joindre des dimensions ordinaires de modèle aux dimensions de mesure, créez des jointures sur les sujets de requête sous-jacents et créez une relation de portée entre la dimension ordinaire et la dimension de mesure.

Lorsque les dimensions reposent sur les métadonnées SAP BW, vous ne pouvez pas modifier la requête sous-jacente.

### Procédure de modification d'une dimension ordinaire

#### Procédure

1. Cliquez sur la dimension ordinaire que vous souhaitez modifier.
2. Dans le menu **Actions**, cliquez sur **Editer la définition**.
3. Sélectionnez l'action voulue :

- Incorporez des calculs en sélectionnant le niveau, en cliquant sur **Ajouter**, puis en définissant l'expression («Création d'un calcul (SAP BW)», à la page 240).
- Incorporez des filtres («Création d'un filtre (SAP BW)», à la page 242).
- Testez la dimension («Test d'une dimension ou d'un autre objet (SAP BW)», à la page 215).

4. Cliquez sur **OK**.

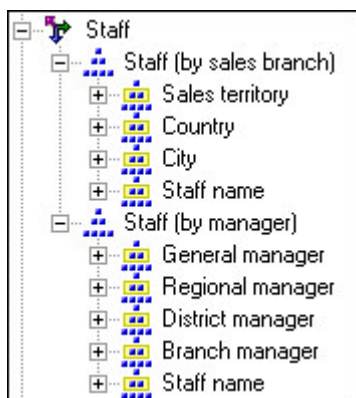
## Hiéarchies pour une dimension ordinaire (SAP BW)

Une hiérarchie est une liste ordonnée de niveaux ou une collection d'éléments. Chaque élément de requête d'une hiérarchie doit porter un nom unique.

Dans IBM Cognos Framework Manager, vous pouvez indiquer plusieurs hiérarchies pour des dimensions ordinaires. Plusieurs hiérarchies définies pour une dimension ordinaire se comportent comme des vues d'une même requête. La première hiérarchie est la hiérarchie par défaut ou principale.

Vous ne pouvez utiliser qu'une seule hiérarchie à la fois dans une requête. Par exemple, vous ne pouvez pas utiliser une hiérarchie sur les lignes d'un rapport de type tableau croisé et une autre hiérarchie de la même dimension dans les colonnes. Si vous avez besoin de ces deux hiérarchies dans le même rapport, vous devez créer deux dimensions, une pour chaque hiérarchie. Pour plus d'informations, voir «Modélisation de dimensions avec plusieurs hiérarchies», à la page 358.

Par exemple, le sujet de requête Personnel de vente peut être visualisé par responsable ou par zone géographique et vous pouvez le modéliser comme une dimension unique avec deux hiérarchies.



IBM Cognos utilise des paramètres par défaut qui n'échoueront pas pour le type de hiérarchie.

- Pour les dimensions qui représentent des caractéristiques SAP BW, la propriété **Équilibré** prend la valeur **True** et la propriété **Décalé** prend la valeur **False**.
- Pour les dimensions qui représentent des hiérarchies de présentation, la propriété **Équilibré** prend la valeur **False** et la propriété **Décalé** prend la valeur **True**.

Ces paramètres peuvent ne pas refléter les valeurs appropriées car IBM Cognos ne détermine pas la structure hiérarchique réelle. Vous pouvez améliorer les performances des modèles et requêtes SAP BW en modifiant les paramètres de



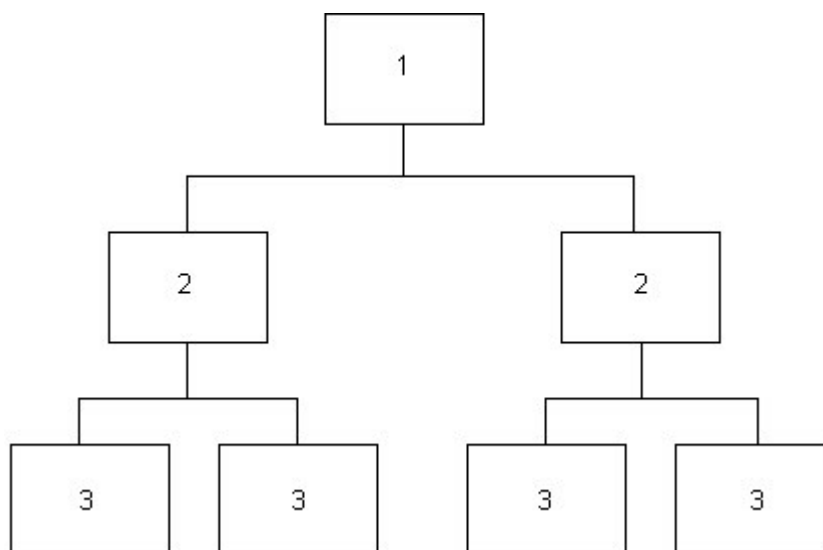
dimension dans IBM Cognos Framework Manager et dans les studios d'IBM Cognos. Pour chaque dimension, vérifiez les propriétés **Équilibré** et **Décalé** afin de vous assurer que les valeurs sont correctement définies.

Outre les hiérarchies dans des dimensions, il existe des hiérarchies dans les métadonnées SAP BW. Pour plus d'informations, voir «Hiérarchies SAP BW», à la page 200.

#### Hiérarchie équilibrée :

Chaque chemin présent dans une hiérarchie équilibrée descend à la même profondeur.

Par exemple, dans le diagramme suivant, le niveau le plus élevé est Ligne de produits (niveau 1), le niveau 2 est Type de produit, puis le niveau 3 est Produits.



Dans SAP BW, tous les noeuds de feuille d'une hiérarchie sont des valeurs de la caractéristique, mais il n'est pas nécessaire que chaque chemin descende au plus bas niveau de la hiérarchie.

Vous pouvez indiquer si une dimension représente une hiérarchie équilibrée en modifiant la propriété **Équilibré** d'une dimension. La valeur que vous définissez dépend du type d'objet représenté par la dimension et selon que la hiérarchie est équilibrée ou non.

Objet représenté par la dimension	Valeur de propriété Équilibré
Caractéristique	true
Hiérarchie de présentation équilibrée	true
Hiérarchie de présentation non équilibrée	false
Hiérarchie de présentation dont la structure est inconnue	false

Pour une dimension qui représente une caractéristique sans une hiérarchie de présentation, cette propriété est accessible en lecture seule et la valeur **true** lui est affectée.

Si une hiérarchie de présentation est équilibrée, affectez la valeur **true** à la propriété **Equilibré** de la dimension qui lui est associée. Par défaut, cette propriété a la valeur **false** pour toutes les hiérarchies de présentation. Une hiérarchie est équilibrée si toutes les valeurs de caractéristique de feuille se trouvent au niveau le plus bas de la hiérarchie. Lorsque la valeur **true** est affectée à la propriété **Equilibré** si cela est nécessaire, le serveur IBM Cognos BI peut générer un code MDX plus efficace.

Lorsque tous les chemins sont de la même profondeur, affectez la valeur **true** à la propriété **Equilibré**, sinon, attribuez-lui la valeur **false**.

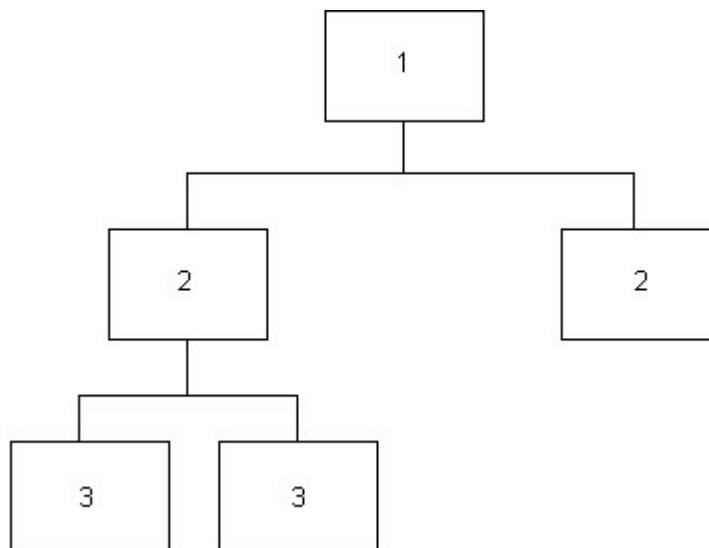
Si vous affectez la valeur **true** à la propriété **Equilibré** d'une hiérarchie non équilibrée, les requêtes qui impliquent cette dimension peuvent renvoyer des données incorrectes.

Si vous affectez la valeur **false** à la propriété **Equilibré** d'une hiérarchie équilibrée, les performances peuvent être ralenties.

#### Hiérarchie non équilibrée :

Les branches d'une hiérarchie non équilibrée ont des niveaux de détail différents.

Par exemple, dans le diagramme suivant, le niveau le plus élevé d'une organisation est le PDG (niveau 1) et le niveau 2 est représenté par les vice-présidents et l'adjoint de direction du PDG. L'adjoint de direction n'a pas subordonnés, contrairement aux vice-présidents.



Une hiérarchie non équilibrée peut également être irrégulière. Dans une hiérarchie non équilibrée irrégulière, les niveaux présentent des trous et ont des niveaux de détail différents.

Dans SAP BW, cela se produit uniquement lorsqu'il y a des nœuds " Non affecté" (ou "#") dans une hiérarchie de présentation. Cependant, la présence d'un tel

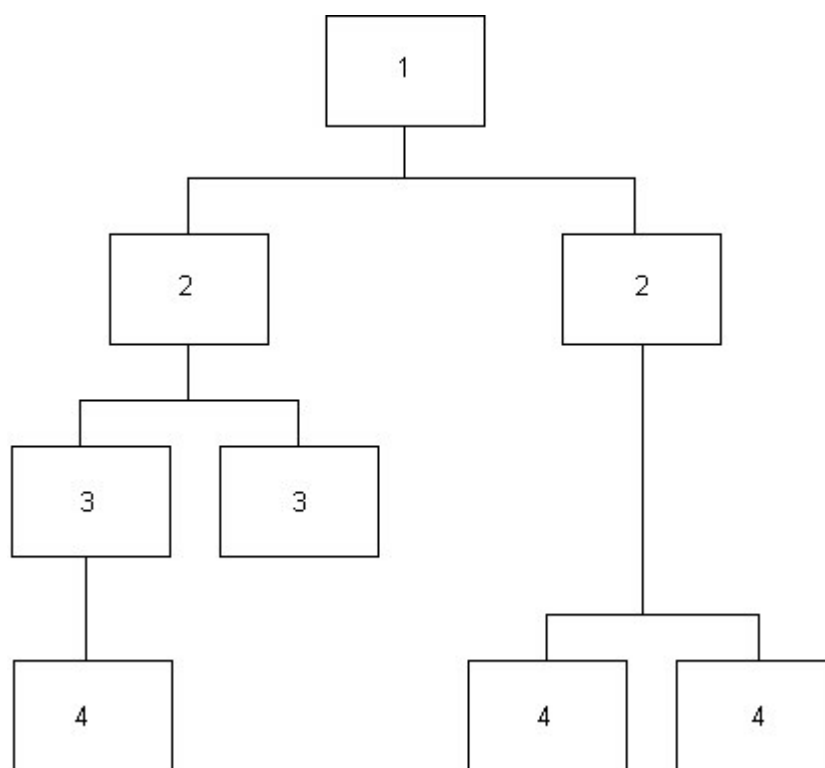
noeud ne suffit pas à rendre la hiérarchie non équilibrée. Vous devez examiner l'agencement d'une hiérarchie pour en être sûr.

Une hiérarchie non équilibrée peut également être irrégulière. Dans une hiérarchie non équilibrée irrégulière, les niveaux présentent des trous et ont des niveaux de détail différents.

#### Hiérarchie irrégulière :

Au moins un chemin dans la hiérarchie ignore au moins un niveau.

Par exemple, le niveau le plus élevé est Société (niveau 1), le niveau 2 est Succursale, le niveau 3 est Bâtiment et le niveau 4 est Département. Certaines succursales peuvent n'avoir qu'un seul bâtiment, avec les mêmes départements que les sites à plusieurs bâtiments.



Dans SAP BW, cela se produit uniquement lorsqu'il y a des noeuds " Non affecté" (ou #) dans une hiérarchie de présentation. Cependant, la présence d'un tel noeud ne suffit pas à rendre la hiérarchie irrégulière. Vous devez examiner l'agencement d'une hiérarchie pour en être sûr.

IBM Cognos utilise des paramètres par défaut qui n'échoueront pas pour le type de hiérarchie.

- Pour les dimensions qui représentent des caractéristiques SAP BW, la propriété **Équilibré** prend la valeur **True** et la propriété **Décalé** prend la valeur **False**.
- Pour les dimensions qui représentent des hiérarchies de présentation, la propriété **Équilibré** prend la valeur **False** et la propriété **Décalé** prend la valeur **True**.

Vous pouvez indiquer si une dimension représente une hiérarchie irrégulière en modifiant la propriété **Décalé** d'une dimension. La valeur que vous définissez dépend du type d'objet que la dimension représente et du fait que vous sachiez si la hiérarchie est irrégulière.

La dimension représente	Valeur de la propriété Décalé
une caractéristique	false
une hiérarchie de présentation qui n'est pas irrégulière	false
une hiérarchie de présentation qui est irrégulière	true
une hiérarchie de présentation dont la structure est inconnue	true

Une dimension qui représente une caractéristique sans hiérarchie de présentation est en lecture seule.

Si une hiérarchie de présentation n'est pas irrégulière, définissez la propriété **Décalé** de la dimension qui lui est associée sur **false**. En définissant la propriété **Décalé** sur **false** selon le cas, le serveur IBM Cognos BI peut générer un langage MDX (Multidimensional Expression Language) plus efficace

Si vous affectez la valeur **true** à la propriété **Décalé** d'une hiérarchie irrégulière, les requêtes qui impliquent cette dimension peuvent renvoyer des données incorrectes.

Si vous affectez la valeur **false** à la propriété **Décalé** d'une hiérarchie irrégulière, les performances risquent d'être ralenties.

#### **Limites relatives aux hiérarchies décalées et non équilibrées et aux valeurs agrégées :**

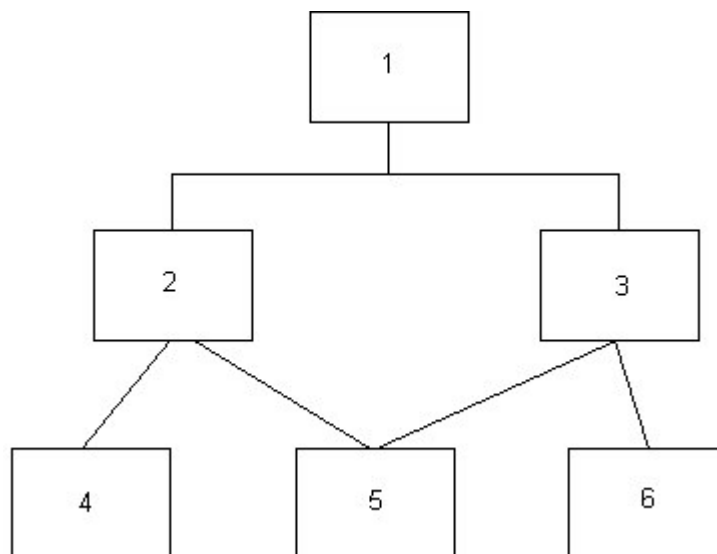
Des hiérarchies non équilibrées et décalées peuvent créer des écarts dans les chemins individuels d'une hiérarchie, ainsi que des noeuds à des niveaux intermédiaires sans descendants aux niveaux inférieurs. Si un rapport contient des éléments de requête d'une seule dimension qui proviennent de niveaux consécutifs ou adjacents dans une hiérarchie, les valeurs de faits associées aux niveaux inférieurs doivent toujours être agrégées aux valeurs associées aux niveaux supérieurs de la dimension dans le rapport.

Toutefois, si les éléments de requête ne proviennent pas de niveaux consécutifs ou adjacents, et si la hiérarchie SAP BW sous-jacente est décalée ou non équilibrée, il est possible que les valeurs des niveaux supérieurs ne répercutent pas l'agrégation des valeurs de fait des niveaux inférieurs. Les valeurs de fait associées aux niveaux supérieurs répercutent les valeurs agrégées de la source de données. Il s'agit d'un comportement classique des sources de données OLAP, mais qui peut paraître non intuitif aux utilisateurs habitués à générer des rapports sur des sources de données relationnelles.

#### **Hiérarchie de réseau :**

Un membre de la hiérarchie possède plus d'un parent.

Par exemple, un employé peut générer des rapports via différentes organisations et avoir plusieurs responsables. Pour SAP BW, cet employé sera pris en compte une seule fois dans le comptage de tous les employés et ne sera pas inclus dans chaque organisation.



### Niveaux pour une dimension ordinaire (SAP BW)

Un niveau est une collection d'attributs avec généralement une granularité commune. Chaque niveau nécessite un élément défini comme clé et un autre élément défini comme légende. Pour les sources de données SAP BW, les niveaux contiennent des membres.

Le premier niveau de la hiérarchie est automatiquement défini en tant que niveau Tout. Il contient un membre racine unique qui représente le niveau supérieur de la hiérarchie. Par exemple, le niveau Tout pour la dimension Temps s'appelle Temps (Tout). Vous ne pouvez pas supprimer ou déplacer le niveau Tout. Par contre, vous pouvez modifier son nom, sa description et son infobulle.

Si vous ne spécifiez pas correctement les niveaux de la hiérarchie, une agrégation incorrecte risque de se produire.

#### Noms uniques de membres :

Le nom unique de membre permet d'identifier un membre dans la source de données, tout comme les clés métier qui permettent de rechercher des enregistrements dans une table.

Le nom unique de membre est utilisé dans l'expression d'un élément de données de membre inclus dans un rapport et dans une référence à des membres dans des filtres et dans des expressions. En outre, il est utilisé lors des opérations d'accès aux détails entre les sources de données OLAP. Les clés de membre figurant dans le nom unique de membre des différentes sources de données OLAP doivent correspondre.

### Rôles (SAP BW)

Vous pouvez également créer des expressions qui font référence à des rôles au lieu de créer des éléments de requête. Vous devez utiliser la fonction `roleValue` pour

faire référence à un rôle particulier. Par exemple, vous souhaitez exécuter une requête sur un rôle spécifique dans une hiérarchie, mais l'élément de requête assumant ce rôle est différent à chaque niveau de la hiérarchie. Une requête unique peut s'appliquer aux différents éléments de requête à chaque niveau. Vous pouvez également utiliser la fonction `roleValue` lorsque vous connaissez le rôle mais pas l'élément de requête sous-jacente.

Vous pouvez affecter plusieurs rôles à un élément de requête, mais le rôle même ne peut pas être attribué à des éléments de requête différents dans le même niveau.

Des rôles par défaut sont prédéfinis pour toutes les hiérarchies parent-enfant et pour tous les niveaux des hiérarchies basées sur des niveaux. La plupart des rôles ne sont pas visibles dans les studios IBM Cognos.

Les rôles qui sont réservés par IBM Cognos BI commencent par un trait de soulignement. Le nom d'un rôle personnalisé ne peut donc pas commencer par un trait de soulignement.

## Rôles par défaut

Les rôles par défaut sont les suivants :

- `_businessKey`  
Représente la clé pour le niveau. Ce rôle est également utilisé pour accéder au détail d'une source de données vers une autre, car la clé métier doit être cohérente dans l'ensemble de votre organisation.  
Le rôle `_businessKey` ne peut être affecté qu'à un seul attribut dans un niveau.
- `_dimensionUniqueName`  
Renvoie le nom de la dimension tel que défini dans le modèle Framework Manager.
- `_hierarchyUniqueName`  
Renvoie le nom de la hiérarchie tel que défini dans le modèle Framework Manager.
- `_levelLabel`  
Renvoie le libellé affecté au niveau.
- `_levelNumber`  
Renvoie le numéro du niveau.
- `_levelUniqueName`  
Renvoie le nom affecté au niveau.
- `_longName`  
Affecté à l'élément de requête qui représente le nom long d'un niveau.
- `_memberCaption`  
Présente la légende pour un membre qui s'affichera dans les studios IBM Cognos.
- `_memberDescription`  
Renvoie la description d'un membre au sein d'une dimension.
- `_memberUniqueName`  
Renvoie le nom unique du membre IBM Cognos.
- `_parentUniqueName`  
Définit le nom affecté au parent de l'élément de requête sélectionné.

- `_planningDefault`  
Indique l'élément de requête à utiliser lorsque des mesures sont sélectionnées. Ce rôle est applicable uniquement à IBM Cognos Planning et SAP BW.
- `_rollupType`  
Définit la manière dont un élément de requête est agrégé.
- `_shortName`  
Affecté à l'élément de requête qui représente le nom abrégé d'un niveau.

Si un élément de requête utilise un rôle par défaut, vous ne pouvez pas modifier son rôle. Ceci s'applique uniquement aux sources de données SAP BW.

## Rôles personnalisés

Vous pouvez créer des rôles personnalisés. Chaque rôle que vous créez doit porter un nom unique. Les rôles qui sont réservés par IBM Cognos commencent par un trait de soulignement. Le nom d'un rôle personnalisé ne peut donc pas commencer par un trait de soulignement. Vous pouvez traduire les rôles personnalisés dans le modèle.

Des rôles portant le même nom que l'élément de requête sont affectés aux propriétés définies par l'utilisateur dans les sources de données OLAP.

### Spécification de rôles (SAP BW)

Les rôles définissent ce qui s'affiche dans l'arborescence de membres dans les studios IBM Cognos. Utilisez des rôles pour organiser et gérer les métadonnées, et déterminer comment présenter les données aux utilisateurs.

### Procédure

1. Cliquez sur la dimension pour laquelle vous souhaitez définir des rôles.
2. Dans le menu **Actions**, cliquez sur **Editer la définition**.
3. Cliquez sur l'onglet **Dimension**.
4. Dans la zone **Hiérarchies**, cliquez sur le niveau souhaité.
5. Dans la zone **Sélectionner un niveau dans le contrôle de hiérarchie afin de voir les éléments de requête**, cliquez sur un élément de requête.
6. Sous **Rôle**, cliquez sur les points de suspension (...).
7. Cliquez sur l'onglet **Rôles personnalisés**, puis sur **Ajouter**.
8. Cliquez sur **Fermer**.
9. Cliquez sur **OK**.

### Résultats

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour définir des rôles. Cliquez sur **Attributs**, cliquez avec le bouton droit de la souris sur l'élément de requête et cliquez sur **Editer les rôles**.

## Modification d'une dimension de chiffres clés (SAP BW)

Une dimension de chiffres clés est une collection de faits, par exemple, Quantité vendue ou Prix.

les chiffres clés sont reliés par le biais des dimensions régulières. Lorsqu'elle est utilisée dans un rapport ou une analyse, la dimension de chiffres clés affiche la valeur de l'élément de requête (par exemple, un nom ou un nombre), ou une

valeur NULL, un zéro ou Non valide. Pour créer des rapports qui comparent et contrastent complètement des zones fonctionnelles, vous pouvez avoir besoin de créer plusieurs dimensions de chiffres clés.

Vous pouvez ajouter de la valeur en intégrant des calculs basés sur des règles métier existants, tels que la marge bénéficiaire.

Vous pouvez également modifier les propriétés de plusieurs dimensions en même temps («Remplacement de plusieurs valeurs de propriété», à la page 17).

Vous ne pouvez pas définir des hiérarchies ou des niveaux pour une dimension de chiffres clés.

Lorsque les dimensions reposent sur les métadonnées SAP BW, vous ne pouvez pas modifier la requête sous-jacente. Vous pouvez cependant ajouter des calculs et des filtres.

### Procédure

1. Cliquez sur la dimension de chiffres clés que vous souhaitez modifier.
2. Dans le menu **Actions**, cliquez sur **Editer la définition**.
3. Sélectionnez l'action voulue :
  - Incorporez des calculs en cliquant sur **Ajouter**, puis en définissant l'expression («Création d'un calcul (SAP BW)», à la page 240).
  - Changez l'ordre des mesures, des éléments ou des calculs en utilisant les boutons flèche. Toutefois, si la dimension de mesure contient un dossier, vous pouvez changer l'ordre des éléments uniquement dans l'**afficheur de projet**.
  - Incorporez des filtres («Création d'un filtre (SAP BW)», à la page 242).
  - Testez la dimension («Test d'une dimension ou d'un autre objet (SAP BW)», à la page 215).
4. Cliquez sur **OK**.

## Affichage d'objets associés (SAP BW)

Vous pouvez explorer une représentation visuelle des objets qui sont connectés au sujet de requête ou à la dimension que vous sélectionnez dans l'**afficheur de projet**. L'explorateur de contexte **Context Explorer** affiche les objets auxquels l'objet sélectionné est connecté. Vous pouvez également sélectionner un objet connecté et voir ses références.




Vous pouvez masquer un objet dans **Context Explorer**. Vous pouvez aussi modifier la présentation, ajuster tous les objets dans **Context Explorer**, effectuer un zoom avant ou arrière, imprimer, afficher un aperçu des diagrammes avant impression et modifier la mise en page.

Vous pouvez également utiliser l'onglet **Grille des dimensions** pour explorer les dimensions.

### Procédure

1. Sélectionnez un ou plusieurs objets à explorer.
2. Dans le menu **Outils**, cliquez sur **Lancer Context Explorer**.
3. Pour afficher les objets connectés, cliquez sur un ou plusieurs objets, puis sur le bouton approprié.



Objectif	Bouton
Afficher les objets qui sont associés à l'objet sélectionné.	
Afficher les références immédiates pour les objets.	
Afficher toutes les références pour les objets.	

- Si vous souhaitez afficher des détails sur un objet, par exemple, ses relations et ses éléments de requête, cliquez avec le bouton droit de la souris sur l'objet, cliquez sur **Parcourir le diagramme**, **Paramètres de diagramme**, puis sélectionnez les détails voulus.

## Test d'une dimension ou d'un autre objet (SAP BW)

Vous pouvez tester une dimension, un niveau, une hiérarchie, des éléments de requête ou un sujet de requête de modèle.

Le test d'une dimension ordinaire renvoie les attributs associés à la première hiérarchie rencontrée dans la dimension.

### Test d'objets

Vous pouvez voir les résultats renvoyés par un objet en le testant. Vous pouvez tester un objet lorsque vous le créez ou ultérieurement. Les objets que vous pouvez tester sont les dimensions, les sujets de requête, les ensembles de requêtes, les hiérarchies, les niveaux, les calculs, et les éléments de requête.

Vous pouvez visualiser les données qui seront affichées dans un rapport spécifique avant de publier un pack en sélectionnant et testant les objets qui apparaîtront dans le rapport. Il est ainsi plus facile de déboguer un modèle et de vérifier que celui-ci répond aux exigences des rapports, car vous n'avez pas besoin de créer et de publier d'abord des packs.

Lorsque vous testez un objet, IBM Cognos Framework Manager renvoie des exemples de données. Aucun formatage n'est appliqué aux exemples de données. Si vous devez tester un formatage, vous devez publier le pack et afficher les objets dans les studios IBM Cognos.

Vous pouvez voir différents résultats en fonction de ce que vous testez. Par exemple, si vous utilisez l'éditeur d'expression pour tester un calcul qui est intégré dans un sujet de requête, Framework Manager teste uniquement l'expression, pas l'élément. Le paramètre d'agrégation pour l'élément de requête n'est donc pas appliqué au test. Le test de la totalité du sujet de requête, qui inclut le calcul, donne un résultat différent car le paramètre d'agrégation est appliqué. Par exemple, si le paramètre d'agrégation a pour valeur Récapituler, vous verrez moins de lignes dans le test.

Lorsque vous testez une dimension de mesure, SQL utilise des agrégats, pas les mesures.

Si vous testez un segment enfant d'un modèle segmenté, une erreur peut s'afficher si un objet que vous testez fait référence à un objet d'un autre segment enfant et que l'objet référencé n'est pas disponible pour le projet en cours. Vérifiez que le modèle parent contient tous les objets et que ce message d'erreur ne s'affiche pas lorsque vous testez le modèle parent.

Des paramètres de restriction peuvent affecter les résultats de test. Pour plus d'informations, voir «Restrictions», à la page 309.

Vous pouvez modifier les paramètres de test existants pour personnaliser les résultats affichés par le test. Par exemple, en plus d'autres paramètres, vous pouvez contrôler le nombre de lignes renvoyées.

## Étapes lors de la création ou de la modification de l'objet (SAP BW)

### Procédure

1. Sélectionnez l'objet à tester.
2. Dans le menu **Actions**, cliquez sur **Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.  
La zone **Résultats du test** est initialement vide tant que vous n'avez pas exécuté la requête.  
Tous les ensembles de résultats qui contiennent des objets BLOB (Binary Large Object) sont affichés en tant que [blob].
3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Si vous testez une expression et que vous souhaitez appliquer la propriété **Agrégat ordinaire** de l'élément de requête ou de la mesure qui est référencé dans l'expression, cochez la case **Total automatique**.  
Si vous désactivez cette case, une ligne est renvoyée pour chaque ligne de l'ensemble de résultats de la requête.
5. Si vous souhaitez obtenir plus d'informations sur les résultats de requête, cliquez sur l'onglet **Informations sur la requête**.
6. Cliquez sur **OK**.

## Procédure de visualisation des données qui seront affichées dans un rapport spécifique

### Procédure

1. Sélectionnez les objets qui s'afficheront dans le rapport.
2. Cliquez sur **Outils, Test**.
3. Pour exécuter la requête et récupérer tous les résultats de test, cliquez sur **Tester l'exemple**.
4. Pour afficher des détails sur tout problème trouvé, cliquez sur l'onglet **Informations sur la requête**.  
Si vous ne voyez pas les résultats de la requête dans la fenêtre de test, il se peut que les données de votre source de données dépassent la valeur de l'une des restrictions. La requête s'arrête à la limite spécifiée, mais la fenêtre des résultats de test ne contient aucune donnée. **Astuce** : Définissez chaque restriction sur zéro.

## Modification des paramètres de test (SAP BW)

Vous pouvez personnaliser les tests en modifiant les paramètres de test.

### Procédure

1. Sélectionnez l'objet de votre choix.
2. Cliquez sur **Actions, Editer la définition**, puis cliquez sur l'onglet **Test** ou **Informations sur la requête**.

3. Cliquez sur **Options, Tester les paramètres.**

4. Sélectionnez les options de votre choix.

Objectif	Action	Persistance
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes.</p> <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>	<p>Ce paramètre s'applique à toutes les dimensions, tous les sujets de requête et tous les ensembles de requêtes du modèle.</p> <p>Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.</p>
Indiquer le niveau de détail	Faites glisser le curseur de <b>Niveau de détail affiché dans les informations de résultat</b> pour représenter la quantité d'informations dont vous avez besoin.	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.
Remplacer temporairement des paramètres de session	<p>Dans la zone <b>Paramètres de session</b>, cliquez sur <b>Définir</b>.</p> <p>La boîte de dialogue <b>Paramètres de session</b> s'affiche.</p>	Ces valeurs de remplacement ne sont pas enregistrées avec le modèle. Ce paramètre est disponible uniquement pour la session en cours.
Appliquer les filtres de mode de conception appropriés	<p>Cochez la case <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b>.</p> <p>Ce paramètre s'applique à tous les filtres pertinents dont l'utilisation est définie sur le mode de conception dans une autre dimension, un autre sujet de requête ou un autre ensemble de requêtes.</p>	Ce paramètre est enregistré et utilisé dans votre prochaine session avec n'importe quel modèle.
Appliquer un filtre de sécurité	Dans la zone <b>Filtres de sécurité</b> , cliquez sur <b>Editer</b> .	Ce paramètre est enregistré et utilisé dans votre prochaine session avec ce modèle.

Objectif	Action	Persistance
Modifier les valeurs d'invites	<p>Dans la zone <b>Valeurs d'invite en cours</b>, cliquez sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>	<p>Les valeurs d'invite ne sont pas enregistrées avec le modèle.</p> <p>Ce paramètre est disponible uniquement pour la session en cours.</p>

5. Cliquez sur le bouton **OK** à deux reprises.

### Résultats

Pour plus d'informations, voir «Dimensions (SAP BW)», à la page 204 et «Sujets de requête de modèle (SAP BW)».

## Sujets de requête de modèle (SAP BW)

Un sujet de requête est un ensemble d'éléments de requête liés par une relation intrinsèque.

Vous pouvez utiliser IBM Cognos Framework Manager pour modifier les sujets de requête afin d'optimiser et de personnaliser les données qu'ils extraient. Par exemple, vous pouvez ajouter des filtres ou des calculs. Lorsque vous modifiez la définition d'un sujet de requête, Framework Manager régénère les éléments de requête associés, pour faire en sorte que toutes les modifications apportées aux propriétés du sujet de requête soient répercutées dans tous les éléments de requête pour ce sujet de requête.

Pour les métadonnées SAP BW, vous pouvez utiliser des sujets de requête de modèle d'IBM Cognos Framework Manager.

Les sujets de requête de modèle ne sont pas générés directement à partir d'une source de données, mais ils sont basés sur des éléments de requête d'autres sujets de requête ou dimensions, y compris des sujets de requête d'autres modèles. En utilisant des sujets de requête de modèle, vous pouvez créer une vue orientée métier plus abstraite d'une source de données.

Si vous créez un sujet de requête de modèle contenant plusieurs dimensions, des problèmes peuvent se produire lorsque vous utilisez le sujet de requête de modèle conjointement avec d'autres sujets de requête ou d'autres dimensions. Assurez-vous que les éléments d'un sujet de requête de modèle ne transgressent pas la logique du modèle, par exemple, l'élément produit inséré entre les éléments pays ou région et ville. Testez le sujet de requête de modèle dans un rapport ; si le groupement fonctionne, le sujet de requête de modèle est valide.

### Création de sujets de requête de modèle (SAP BW)

Vous pouvez également créer un sujet de requête de modèle en fusionnant des sujets de requête et des éléments de requête existants («Création de sujets de requête de modèle basés sur des objets existants (SAP BW)», à la page 219).

## Procédure

1. Sélectionnez le dossier des espaces-noms et, dans le menu **Actions**, cliquez sur **Créer, Sujet de requête**.
2. Dans la zone **Nom**, entrez un nom pour le nouveau sujet de requête.
3. Cliquez sur **Modèle**, puis sur **OK**.

**Remarque :** Pour les métadonnées SAP BW, vous pouvez créer uniquement des sujets de requête de modèle.

4. Cliquez sur l'onglet **Définition de sujet de requête**.
5. Pour ajouter des éléments à un sujet de requête de modèle, faites glisser les éléments depuis la zone **Objets de modèle disponibles** vers la zone **Éléments de requête et calculs**.

Vous pouvez changer l'ordre des éléments et des calculs. Toutefois, si le sujet de requête contient un dossier d'élément de requête, vous pouvez changer l'ordre des éléments uniquement dans l'**afficheur de projet**.

6. Pour incorporer des calculs dans le sujet de requête de modèle, cliquez sur **Ajouter** et définissez le calcul.
7. Pour incorporer des filtres dans le sujet de requête de modèle, cliquez sur l'onglet **Filtres**.
8. Pour tester le sujet de requête de modèle, cliquez sur l'onglet **Test**.
9. Cliquez sur **OK**.

Un avertissement s'affiche si des modifications ont invalidé des relations, d'autres sujets de requête, des calculs ou des filtres.

## Résultats

Vous pouvez être amené à consulter les rubriques associées suivantes :

- calculs intégré : «Création d'un calcul (SAP BW)», à la page 240
- filtres intégrés : «Création d'un filtre (SAP BW)», à la page 242
- test et définition des options de test : «Test d'une dimension ou d'un autre objet (SAP BW)», à la page 215
- modification des propriétés pour plusieurs sujets de requête en même temps : «Remplacement de plusieurs valeurs de propriété», à la page 17

## Création de sujets de requête de modèle basés sur des objets existants (SAP BW)

Vous pouvez sélectionner des objets de modèle existants et les fusionner dans un nouveau sujet de requête de modèle.

Vous pouvez ainsi réutiliser des métadonnées existantes pour créer rapidement des sujets de requête.

Vous pouvez notamment fusionner les objets suivants :

- Sujets de requête de modèle et leurs raccourcis.
- Éléments de requête, filtres et calculs dans des sujets de requête de modèle.

Vous pouvez fusionner un nombre quelconque d'objets de même type dans une nouvelle requête en une seule opération. La fusion crée toujours un nouveau sujet de requête de modèle.

Le nouveau sujet de requête contient les filtres qui existaient dans le sujet de requête d'origine.

Vérifiez que les sujets de requête de modèle ne transgressent pas la logique du modèle. Par exemple, si un sujet de requête avec plusieurs caractéristiques est utilisé en combinaison avec d'autres sujets de requête, des problèmes peuvent survenir lorsque vous exécutez le rapport.

N'incluez d'éléments de requête issus de sujets de requête différents ou de hiérarchies de la même dimension. Cela entraîne une erreur d'exécution.

### Procédure

1. Appuyez sur Ctrl et cliquez sur les objets à fusionner dans un sujet de requête unique.
2. Cliquez sur **Actions, Fusionner dans le nouveau sujet de requête.**

### Validation de sujets de requête de modèle (SAP BW)

Vous pouvez valider la définition du sujet de requête sans ouvrir la boîte de dialogue **Définition de sujet de requête**. Cela s'avère utile lorsque

- De nouveaux éléments de requête sont ajoutés à un sujet de requête
- La définition du sujet de requête sous-jacent a été modifiée

La commande **Valider l'objet** évalue les objets sélectionnés et s'assure qu'ils peuvent s'exécuter.

Lorsqu'IBM Cognos Framework Manager évalue un sujet de requête, une demande est envoyée à la source de données SAP BW. Les attributs physiques, comme le type de données, sont mis à jour selon les besoins pour les sujets de requête.

Vous pouvez également synchroniser la totalité du projet («Synchronisation des projets», à la page 305).

### Procédure

1. Sélectionnez le sujet de requête à évaluer.
2. Dans le menu **Outils**, cliquez sur **Valider l'objet**.

### Résultats

Si vous avez modifié la propriété **Agrégat ordinaire** sur **Non pris en charge**, la propriété est réinitialisée lorsque vous évaluez le sujet de requête. Si la propriété est définie sur une autre valeur, elle n'est pas modifiée.

**Remarque :** Un message d'erreur s'affiche pour chaque sujet de requête non valide. L'objet est également associé au statut Non valide.

---

## Éléments de requête (SAP BW)

Un élément de requête est le plus petit élément du modèle qui peut être placé dans un rapport. Il représente une instance unique d'un élément, telle que la date à laquelle un produit a été mis sur le marché.

Les chiffres clés et les attributs sont importés en tant qu'éléments de requête dans IBM Cognos Framework Manager.

Une seule hiérarchie d'une dimension doit être utilisée dans un même rapport.

Pour les métadonnées SAP BW, vous pouvez modifier uniquement les propriétés de texte, telles que le nom ou l'infobulle.

Comme les rapports peuvent contenir des éléments de requête différents issus d'un ou de plusieurs objets du modèle, les propriétés d'élément de requête contrôlent de nombreux aspects du rapport final. Lorsque vous créez une dimension de modèle ou un sujet de requête de modèle, les éléments de requête héritent des propriétés des éléments de requête de source de données sur lesquels ils sont basés.

Les propriétés des éléments de requête ou des mesures sont notamment les suivantes :

Propriété d'élément de requête	Description
Nom	Nom de l'élément de requête ou de la mesure.
Description	Description de l'élément de requête ou de la mesure.
Dernière modification	Date à laquelle l'élément de requête ou la mesure a été modifié pour la dernière fois. La propriété est automatiquement mise à jour avec l'heure et la date en cours.
Dernière modification effectuée par	Utilisateur qui a modifié en dernier l'élément de requête ou la mesure. Cette propriété est automatiquement mise à jour lorsque l'élément est modifié.  Sa valeur est le nom d'utilisateur de connexion en cours.
Commentaires de modèle	Permet d'ajouter des commentaires internes concernant le modèle. Ces informations sont utilisées dans la boîte de dialogue <b>Analyser l'impact de publication</b> et dans le <b>rapport de modèle</b> . Les commentaires ne sont pas accessibles aux utilisateurs du pack.
Infobulle	Description pouvant s'afficher dans le pack publié pour les utilisateurs.
Expression	Permet de créer des calculs intégrés qui fournissent aux utilisateurs les valeurs calculées qu'ils utilisent régulièrement.  Cette propriété est destinée aux mesures uniquement.  <b>Remarque :</b> La propriété <b>Expression</b> n'est pas utilisée par SAP BW.
Nom externe	Nom qui s'affiche dans la source de données.

Propriété d'élément de requête	Description
Est masqué	<p>Indique s'il faut masquer ou afficher l'élément de requête ou la mesure dans le pack publié.</p> <p>Même si <b>Est masqué</b> est défini sur <b>True</b> et que l'élément de requête ou la mesure est invisible pour les utilisateurs, l'élément de requête ou la mesure figure quand-même dans le pack publié car ils peuvent être requis par d'autres objets du modèle. Vous ne voyez pas l'élément de requête ou la mesure dans l'assistant de <b>publication de pack</b>.</p> <p>Par exemple, un calcul peut utiliser un élément de requête masqué.</p>
Usage	<p>Utilisation prévue pour les données représentées par l'élément de requête.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p>
Format	<p>Manière dont les informations apparaissent dans un rapport.</p>
Devise	<p>Devise utilisée.</p> <p>Cette propriété ne peut pas être modifiée dans la sous-fenêtre <b>Propriétés</b>. Utilisez la propriété <b>Format</b> pour modifier la devise.</p>
Type de données	<p>Type de données défini dans la source de données.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Précision	<p>Nombre total de chiffres.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Echelle	<p>Nombre de chiffres représentés dans l'échelle.</p> <p>Par exemple, vous pouvez afficher des nombres en milliers de sorte que 100 000 corresponde à 100 000 000.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>



Propriété d'élément de requête	Description
Taille	<p>Taille de l'élément de requête ou de la mesure.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
NULL admis	<p>Indique si l'élément de requête ou la mesure peut contenir une valeur NULL.</p> <p>Comme cette propriété est définie dans la source de données, elle est en lecture seule dans Framework Manager.</p>
Type d'affichage	<p>Indique comment l'élément de requête est affiché.</p> <p>La valeur de colonne peut s'afficher dans les studios IBM Cognos sous la forme d'une image, d'un lien ou d'une valeur.</p> <p>La valeur par défaut est value.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p>
Type MIME	<p>Format utilisé par la valeur de colonne.</p> <p>Par exemple, si <b>Type d'affichage</b> est définie sur picture, <b>Type MIME</b> peut être jpeg.</p> <p>Cette propriété est destinée aux éléments de requête uniquement.</p> <p><b>Remarque :</b> La propriété <b>Type MIME</b> propriété n'est pas utilisée par SAP BW.</p>
Informations d'invite	Comportement de l'invite.
Agrégat ordinaire	Type d'agrégation associé à l'élément de requête, la mesure ou le calcul dans le pack publié.
Règles d'agrégation	<p>Pour les métadonnées relationnelles modélisées de façon dimensionnelle, les règles pour l'agrégation semi-additive.</p> <p>Pour les métadonnées SAP BW, la propriété <b>Règles d'agrégation</b> est accessible en lecture seule.</p>

Propriété d'élément de requête	Description
Règle d'allocation	Indique le type d'allocation défini pour la mesure.  La valeur <b>par défaut</b> indique que l'allocation constante est utilisée dans les requêtes de type Liste et l'allocation unique est utilisée dans les requêtes de tableau croisé. La valeur <b>constante</b> indique que l'allocation constante est utilisée dans toutes les requêtes.
N'est pas triable	Indique si les valeurs de cet élément de requête peuvent être triées.  Cette propriété est destinée aux éléments de requête contenant des objets volumineux comme les objets BLOB.

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Propriétés **Utilisation** et **Agrégat ordinaire** «Modification du mode d'agrégation des éléments de requête (SAP BW)»
- Modification du symbole monétaire «Formatage des éléments de requête (SAP BW)», à la page 227
- Invites «Définition d'un contrôle d'invite (SAP BW)», à la page 228
- Modification des propriétés pour plusieurs éléments de requête en même temps «Remplacement de plusieurs valeurs de propriété», à la page 17

## Modification du mode d'agrégation des éléments de requête (SAP BW)

Lorsque vous importez des métadonnées, Framework Manager attribue des valeurs aux propriétés **Usage** et **Agrégat ordinaire** pour les éléments de requête et les mesures en fonction de leur type d'objet. La propriété **Usage** identifie l'utilisation prévue des données représentées par l'élément de requête («Propriété Usage», à la page 140). La propriété **Agrégat ordinaire** identifie le type d'agrégation appliqué à l'élément de requête ou la mesure («Propriété Agrégat ordinaire», à la page 141). Les utilisateurs peuvent redéfinir les valeurs de la propriété **Agrégat ordinaire**. Pour les mesures semi-additives, vous pouvez spécifier des règles d'agrégation supplémentaires en modifiant la propriété **Règles d'agrégation** («Définition de règles d'agrégation pour des mesures semi-additives», à la page 141).

Lorsque vous modifiez la propriété **Agrégat ordinaire**, vous pouvez sélectionner des valeurs qui ne sont pas disponibles via l'importation, par exemple, Moyenne et Maximum. Vous devez comprendre ce que représentent les données pour savoir quelle valeur d'agrégat est nécessaire. Par exemple, si vous agrégez un numéro de composant, les seules valeurs d'agrégat qui s'appliquent sont Nombre, Nombre (éléments distincts), Maximum et Minimum.

En règle générale, préférez l'utilisation de la propriété **Agrégat ordinaire** à l'ajout de fonctions récapitulatives dans les expressions. Utilisez des fonctions

récapitulatives dans les expressions si le récapitulatif requis n'est pas pris en charge par une propriété d'agrégat, ou si l'expression est trop complexe pour être gérée par les propriétés des éléments.

La méthode suivante de comptage des valeurs distinctes, par exemple des clients, garantit la cohérence des résultats dans les listes, les tableaux croisés et les graphiques. Il est recommandé de compter des éléments distincts sur une clé métier qui identifie les valeurs de façon unique plutôt que sur une valeur de chaîne. Cette opération prévient la perte involontaire des lignes uniques contenant un nom identique.

L'exemple suivant compte les clients distincts dans un modèle relationnel. Créez une copie de **Clé client** dans le sujet de requête **Client**, nommez-la **Nombre de clients**, définissez la propriété **Utilisation** sur **Fait** et la propriété **Agrégat ordinaire** sur **Nombre (éléments distincts)**. Il est déconseillé de créer un élément de requête avec l'expression `Count (distinct [Customer Key])`, car le comportement d'un élément de requête peut être imprévisible selon les scénarios. Si un formatage est requis, définissez le **type de format** sur **Nombre** et définissez les propriétés de **format** adéquates, par exemple le nombre de décimales ou le séparateur de milliers.

Dans un modèle DMR (Dimensionally Modeled Relational), ajoutez l'élément de requête **Nombre de clients** créé dans le sujet de requête **Client** à une dimension de mesure. La propriété **Agrégat ordinaire** et les paramètres de la propriété **Format** sont hérités. Une autre solution consiste à ajouter la **Clé client** d'origine du sujet de requête **Client** à la dimension de mesure, à définir la propriété **Agrégat ordinaire** sur **Nombre (éléments distincts)**, et à définir le format nécessaire.

## Règles de définition de propriétés pour des dimensions (SAP BW)

IBM Cognos Framework Manager utilise les règles suivantes pour définir les propriétés **Usage** et **Agrégat ordinaire**.

Objet	Propriété Usage	Propriété Agrégat ordinaire
Élément de requête dans une dimension ordinaire	Attribut	Non pris en charge
Élément de requête dans une dimension de mesure	Identificateur	Nombre
Mesure dans une dimension de mesure	Fait	Automatique si la mesure est un calcul Somme si la mesure n'est pas un calcul

Si la mesure est semi-additive, utilisez la propriété **Règles d'agrégation** pour définir les règles d'agrégation semi-additive. Voir «Définition de règles d'agrégation pour des mesures semi-additives», à la page 141.

Pour les métadonnées SAP BW, vous ne pouvez pas modifier ces propriétés pour des dimensions.

## Règles de définition de propriétés pour des calculs (SAP BW)

La propriété **Agrégat ordinaire** pour un calcul dans des métadonnées SAP BW est définie sur **Automatique**. Pour déterminer la portée d'automatique, les règles suivantes s'appliquent.

Calcul	Type d'agrégation
éléments clés	non pris en charge
tous les autres éléments	calculé

## Règles de définition de propriétés pour des sujets de requête de modèle (SAP BW)

Pour les sujets de requête de modèle, Framework Manager utilise les paramètres de l'objet sur lequel est basé le sujet de requête modèle.

**Remarque :** Si vous modifiez une valeur d'agrégation pour des métadonnées SAP BW, l'agrégation ne peut pas exécuter des requêtes de temps, car les règles d'agrégation ne sont pas appliquées.

## Propriété Usage

Vous devez vérifier si cette propriété est définie correctement. Par exemple, si vous importez une colonne numérique qui participe à une relation, la propriété **Usage** est définie sur **identificateur**. Vous pouvez modifier la propriété.

Pour les éléments de requête SAP BW, la valeur de la propriété **Usage** dépend du type d'élément dimensionnel sur lequel est basée la requête.

Valeur de la propriété Usage	Objet SAP BW	Description
Identificateur	niveau de hiérarchie	Identifie de manière unique les valeurs de caractéristique à un niveau particulier d'une hiérarchie.
Fait	chiffre clé	Représente un chiffre clé, généralement, des données numériques. Les données de date et d'heure sont également prises en charge.
Attribut	attribut d'affichage	Représente un attribut d'affichage qui est associé à une caractéristique.

## Propriété Agrégat ordinaire

Par exemple, si la valeur de la propriété **Agrégat ordinaire** de l'élément de requête Quantité est Somme et que le groupement se fait par nom de produit dans le rapport, la colonne Quantité du rapport montre la quantité totale de chaque produit.

Les types d'agrégation suivants sont pris en charge pour les sources de données SAP BW :

- Automatique
- Moyenne
- Moyenne non nulle

Ce type est pris en charge uniquement lorsqu'il est défini dans la source de données. Vous ne pouvez pas modifier la propriété en Moyenne non nulle dans Framework Manager.

- Calculé
- Nombre
- Nombre (éléments distincts)
- Nombre non nul

Ce type est pris en charge uniquement lorsqu'il est défini dans la source de données. Vous ne pouvez pas modifier la propriété en Nombre non nul dans Framework Manager.

- Maximum
- Médian
- Minimum
- Ecart type
- Somme
- Variance

### Règles permettant de déterminer le type d'agrégation automatique :

Si le calcul est effectué dans un objet SAP BW, ces règles s'appliquent.

Calcul	Type d'agrégation
éléments clés	non pris en charge
tous les autres éléments	calculés

### Propriété Règles d'agrégation

Pour les métadonnées relationnelles modélisées de façon dimensionnelle, cette propriété définit la méthode d'agrégation d'une mesure semi-additive pour les dimensions que vous sélectionnez.

Pour les métadonnées SAP BW, la propriété **Règles d'agrégation** est accessible en lecture seule. La propriété **Semi-agrégat** est utilisée à la place.

### Propriété Semi-agrégat

Pour les métadonnées SAP BW, la propriété **Semi-agrégat** indique la valeur qui est définie dans la source de données. Cette propriété est en lecture seule.

Si la valeur est définie sur **non pris en charge** dans IBM Cognos Framework Manager, le comportement de semi-agrégation est ignoré dans les studios IBM Cognos.

La propriété **Semi-agrégat** ne sera pas prise en charge dans les futures éditions. Utilisez plutôt la propriété **Règles d'agrégation** pour les mesures semi-additives.

## Formatage des éléments de requête (SAP BW)

Par exemple, vous pouvez affecter le type de format **Devise** à un élément de requête numérique, puis utiliser la propriété **Nombre de décimales** de la boîte de dialogue **Format de données** pour indiquer le nombre de positions décimales qui s'affichent dans les rapports.

Certains caractères sont sensibles à la langue et s'affichent correctement uniquement lorsque vos paramètres régionaux prennent en charge la police applicable. Par exemple, pour que le symbole de la devise japonaise s'affiche correctement, la valeur Japonais doit être affectée à vos paramètres régionaux.

Si IBM Cognos Framework Manager n'affiche pas la devise que vous demandez, prenez soin d'installer les modules de langue appropriés pour permettre la prise en charge des symboles de devise. Par exemple, pour que le symbole de la devise indienne (roupie) puisse s'afficher, vous devez exécuter un système d'exploitation ou installer un module de langue qui peuvent afficher ce symbole. Le système d'exploitation japonais ou la langue japonaise permettent d'afficher le symbole de la devise indienne.

Vous pouvez définir des propriétés pour plusieurs éléments de requête en même temps. Toutefois, si les éléments de requête possèdent différents types de format, toutes les propriétés déjà spécifiées sont remplacées et les valeurs par défaut de la source de données sont utilisées. Si les types de format d'origine des éléments de requête sélectionnés sont identiques, toutes les propriétés des éléments de requête sélectionnés sont définies de manière identique.

Par exemple, pour utiliser le même séparateur décimal pour deux éléments de requête et maintenir un nombre de décimales différent, chaque élément de requête doit être modifié individuellement. Si deux éléments de requête sont modifiés en même temps, toutes les propriétés, y compris le nombre de positions décimales, sont définies de manière identique pour ces deux éléments de requête.

## Procédure de formatage des éléments de requête (SAP BW)

### Procédure

1. Dans la sous-fenêtre **Afficheur de projet**, cliquez sur l'élément de requête à modifier.
2. Dans l'onglet **Propriétés** de la sous-fenêtre **Propriétés**, cliquez sur la propriété **Format**.
3. Définissez le type de format sur **Devise** pour que le format de devise soit appliqué à tous les types de rapports.
4. Dans la zone **Portée de la devise**, indiquez le type de devise. Si vous ne voyez pas la devise que vous souhaitez utiliser, cliquez sur le bouton **Ajouter**.
5. Dans la zone **Propriétés**, sélectionnez ou entrez la valeur de propriété appropriée.
6. Cliquez sur **OK**.

## Définition d'un contrôle d'invite (SAP BW)

Vous pouvez utiliser des invites dans :

- des tables de mappage des paramètres ;
- des paramètres de session ;
- des expressions, y compris des filtres et des calculs.

**Remarque :** si la légende est d'un type de données différent de celui de la clé métier pour le niveau, utilisez la propriété **Filtrer la référence d'élément** conjointement avec la valeur **Type d'invite** pour la légende. Cela permet de garantir que le type de données approprié sera utilisé lors du filtrage dans les studios.

Ceci est utile pour les éléments de requête, tels que ProductTypeCode, dont les valeurs ne sont pas affichées dans un rapport, mais sont utiles pour le filtrage des données. En règle générale, il est préférable de définir des invites de saisie au clavier dans les rapports afin de pouvoir utiliser des fonctions d'invite supplémentaires. Cependant, vos utilisateurs ne peuvent pas modifier certaines variables. Pour ces variables, vous pouvez utiliser IBM Cognos Framework Manager au lieu des rapports pour définir des invites de saisie au clavier.

Les propriétés Informations d'invite définies dans Framework Manager vous offrent la possibilité de contrôler le filtrage et les invites par défaut. Les propriétés sont utilisées par :

- Query Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- L'outil Créer une page d'invite de Report Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- Des invites générées dans Report Studio, pour définir l'utilisation et l'affichage d'éléments dans l'invite et la requête d'invite.

La syntaxe permettant d'utiliser une invite en tant que valeur est la suivante :  
?<PromptName>?

## Procédure de définition d'un contrôle d'invite

### Procédure

1. Cliquez sur l'élément de requête.
2. Dans la sous-fenêtre **Propriétés**, cliquez sur l'onglet **Propriétés**.
3. Cliquez sur le signe plus (+) en regard de la propriété **Informations d'invite**.  
Il s'agit d'une propriété d'élément de requête composée.
4. Modifiez les propriétés suivantes selon le comportement dont vous avez besoin.

Objectif	Propriété
Définir le type de contrôle d'invite qui est généré lorsque le rapport est exécuté.	Type d'invite
Définir l'invite générée dans le cadre d'une série d'invites en cascade générées.	Cascade sur la référence d'élément
Indiquer l'élément de requête qui s'affiche pour l'utilisateur du rapport dans l'invite.  Les valeurs de l'invite sont les valeurs de données de l'élément de requête.  Chaque valeur de l'invite est associée à une valeur de l'élément de requête spécifié dans la propriété <b>Utiliser la référence d'élément</b> .	Afficher la référence d'élément
Indiquer l'élément de requête qui est transmis de l'invite au filtre.  Chaque valeur est associée à une valeur de l'élément de requête spécifié dans la propriété <b>Afficher la référence d'élément</b> .	Utiliser la référence d'élément

Objectif	Propriété
Indiquer l'élément de requête utilisé dans l'expression de filtre pour extraire des données.	Filtrer la référence d'élément

### Propriété de type d'invite

La propriété **Type d'invite** définit le type de contrôle d'invite généré lorsque le rapport est exécuté, par exemple, une zone d'édition ou une liste déroulante.

La valeur par défaut de cette propriété est **Déterminé par le serveur**.

**Remarque:** Les types d'invite définis sur des attributs sont désormais traités. L'utilisateur du rapport verra l'invite qui correspond au type d'invite de l'attribut. Comme les types d'invite sur des attributs n'étaient pas traités dans la version précédente, certaines différences peuvent avoir lieu.

Valeur	Contrôle d'invite
Déterminé par le serveur	Le type de contrôle d'invite dépend d'informations sur le serveur, telles que le type de données de l'élément de requête.
Zone d'édition	Zone de texte simple. Si le type de données de la colonne est date ou date-heure, cette valeur génère un contrôle de date ou de date-heure, ainsi que la zone de texte.
Sélectionner une date	Contrôle de date avec une interface de calendrier.
Sélectionner une date/heure	Contrôle de date-heure avec une interface de calendrier.  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.
Sélectionner un intervalle	Contrôle d'intervalle de date-heure  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.
Sélectionner une heure	Contrôle d'heure qui filtre les données en fonction de la période sélectionnée. Par exemple, si vous définissez une invite Sélectionner une heure pour Heure de la commande, l'utilisateur peut utiliser le contrôle d'heure pour afficher toutes les commandes passées après 1:00, ou toutes les commandes passées entre 10:00 et 11:00.  Si vous faites référence à un membre d'heure, vous devez utiliser les valeurs exactes uniquement.  Si vous utilisez une plage, les limites de la plage doivent correspondre à des valeurs de la source de données.
Sélectionner une valeur	Liste déroulante.



Valeur	Contrôle d'invite
Sélectionner avec recherche	Contrôle de liste permettant à des utilisateurs de rechercher des valeurs.  Pour des métadonnées SAP BW, cette valeur n'est pas pertinente.
Sélectionner avec arborescence	Contrôle d'invite hiérarchique pour les invites qui sont basées sur un noeud hiérarchique.

### Propriété Cascade sur la référence d'élément

La propriété **Cascade sur la référence d'élément** indique que l'invite générée fait partie d'une série d'invites en cascade générées. L'élément de requête que vous référencez dans cette propriété est l'élément parent de la cascade. Le système invite l'utilisateur à entrer l'élément en cascade avant l'élément de requête en cours.

Par exemple, si vous souhaitez inviter l'utilisateur à saisir la ligne de produits, puis le produit au sein de la ligne sélectionnée, définissez la propriété **Cascade sur la référence d'élément** de l'élément de requête Produit sur Ligne de produits.

### Propriétés Afficher la référence d'élément et Utiliser la référence d'élément

La propriété **Afficher la référence d'élément** indique quel élément de requête est affiché pour l'utilisateur dans l'invite. La propriété **Utiliser la référence d'élément** indique quel élément de requête est transmis de l'invite au filtre. Chaque valeur de la liste des éléments d'affichage est associée à une valeur de l'élément de requête spécifié dans la propriété **Utiliser la référence d'élément**.

Par exemple, vous souhaitez que l'invite Nom de pays s'affiche lors de l'utilisation de Code de pays pour extraire des données. Affectez la valeur Nom de pays à la propriété **Afficher la référence d'élément** et la valeur Code de pays à la propriété **Utiliser la référence d'élément**. L'invite pour Nom de pays permet à l'utilisateur du rapport de sélectionner plus facilement les valeurs requises. Cependant, l'utilisation de Code de pays dans le filtre s'avère plus efficace pour l'extraction de données.

Ces propriétés sont utilisées par :

- Query Studio, pour créer une expression de filtre et définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- L'outil Créer une page d'invite de Report Studio, pour définir l'utilisation et l'affichage d'éléments dans une invite et une requête d'invite.
- Des invites générées dans Report Studio, pour définir l'utilisation et l'affichage d'éléments dans l'invite et la requête d'invite.

**Remarque :** les valeurs des propriétés **Utiliser la référence d'élément** et **Filtrer la référence d'élément** doivent être compatibles. Sinon, les résultats de l'utilisateur du rapport peuvent être inattendus. Pour plus d'informations, voir «Propriété Filtrer la référence d'élément», à la page 149.

Valeur par défaut : si aucune valeur n'est définie, les propriétés prennent par défaut le nom de l'élément de requête.

Ces propriétés sont utilisées uniquement pour les contrôles d'invite pilotés par des données dont la propriété **Type d'invite** a pour valeur **Sélectionner une valeur** ou **Sélectionner avec recherche**.

### **Propriété Filtrer la référence d'élément**

La propriété **Filtrer la référence d'élément** identifie l'élément de requête utilisé lorsque Report Studio ou Query Studio génère un filtre. Cette propriété facilite la création de requêtes plus efficaces en faisant en sorte qu'un filtre utilise une colonne numérique indexée plutôt qu'une colonne de chaîne non indexée.

Par exemple, un auteur de rapport souhaite créer un filtre pour l'élément de requête Nom de pays. Vous définissez la propriété **Filtrer la référence d'élément** de sorte que l'élément Code de pays soit utilisé au lieu de l'élément Nom de pays dans n'importe quel filtre qui utilise l'élément de requête Nom de pays.

Dans un autre exemple, un auteur de rapport veut créer un filtre pour l'élément de requête Code de pays qui s'affiche dans la table Commandes. Vous souhaitez que ce filtre utilise l'élément Code de pays de la table Pays car celle-ci contient moins de lignes à lire. Par conséquent, vous affectez `Country.Country Code` à la propriété **Filtrer la référence d'élément** dans le modèle.

Cette propriété est utilisée par :

- Query Studio, pour créer une expression de filtre.
- L'outil Créer une page d'invite de Report Studio, pour créer une expression de filtre.

Valeur par défaut : si aucune valeur n'est définie, la propriété prend par défaut le nom de l'élément de requête.

### **Utilisation des propriétés Filtrer la référence d'élément et Utiliser la référence d'élément**

Les valeurs des propriétés **Filtrer la référence d'élément** et **Utiliser la référence d'élément** doivent être compatibles. La valeur de la propriété **Utiliser la référence d'élément** doit correspondre à un type attendu par la propriété **Filtrer la référence d'élément**. Sinon, les résultats de l'utilisateur du rapport peuvent être inattendus. Cela peut se produire lorsqu'un utilisateur de rapport crée un filtre sans créer de page d'invite.

Dans un exemple de modèle, la propriété **Utiliser la référence d'élément** prend la valeur Numéro d'employé et la propriété **Filtrer la référence d'élément** prend la valeur Nom d'employé. Dans Report Studio, un auteur de rapport crée le filtre suivant sans créer de page d'invite :

```
Reference.EmployeeName in ?parm1?
```

Report Studio génère automatiquement des invites lorsque vous créez un filtre sans créer de page d'invite. Etant donné que l'invite est générée, Report Studio utilise les propriétés Informations d'invite de l'élément de requête Nom d'employé dans le modèle Framework Manager.

La propriété **Utiliser la référence d'élément** indique que les valeurs transmises au filtre sont des numéros d'employé. La propriété **Filtrer la référence d'élément** filtre les données en fonction de la valeur Nom d'employé. Le filtre se présente comme suit : `Reference].[Employee Name] in ("1", "2")`. Dans la mesure où il n'existe aucune valeur "1" ou "2" pour le nom d'employé, le rapport est vide.

## Utilisation de la propriété **Filtrer la référence d'élément** pour les métadonnées relationnelles modélisées de façon dimensionnelle

Pour les métadonnées relationnelles modélisées de façon dimensionnelle, la propriété **Informations d'invite** est spécifiée sur l'attribut avec le rôle `_memberCaption`, et non pas sur le niveau. Bien qu'elles soient définies sur l'attribut, les propriétés **Informations d'invite** sont traitées comme si elles étaient spécifiées sur le niveau. Par défaut, lorsque le niveau est inclus dans un rapport, les utilisateurs sont invités à entrer des noms uniques de membre à l'invite de niveau. Pour entrer des valeurs de légende à la place, définissez la propriété **Filtrer la référence d'élément** de l'attribut sur lui-même. Lorsque le filtre obtenu par invite est appliqué, les valeurs filtrées sont basées sur les valeurs d'attributs.

Par exemple, le niveau Ligne de produits possède un attribut Ligne de produits doté d'un rôle `_memberCaption`. Si la valeur de propriété **Filtrer la référence d'élément** a pour valeur Ligne de produits, les utilisateurs de rapport sont invités à entrer des valeurs Ligne de produits. Si la valeur de propriété **Filtrer la référence d'élément** est vide, les utilisateurs sont invités à entrer des noms uniques de membre.

**Remarque** : vous ne devez pas utiliser la propriété **Filtrer la référence d'élément** avec le type d'invite Sélectionner avec arborescence. Etant donné qu'un type d'invite Sélectionner avec arborescence peut uniquement filtrer sur un niveau ou une hiérarchie, le fait de définir la propriété **Filtrer la référence d'élément** va provoquer une erreur.

### Test des invites

Lorsque vous testez un objet de modèle qui fait référence à une invite, IBM Cognos Framework Manager vous demande d'entrer la valeur de l'invite. Framework Manager utilise cette valeur pendant la durée de la session ou jusqu'à ce que vous désélectionniez la valeur de l'invite.

Vous pouvez modifier la valeur de session des valeurs d'invite par le biais de la boîte de dialogue **Options**. Cette boîte de dialogue est accessible lorsque vous modifiez une dimension ou un sujet de requête, ou que vous définissez un calcul, un filtre, un ensemble de requêtes ou une relation complexe. Vous pouvez modifier la valeur d'invite lors du test de l'expression qui fait référence à cette valeur.

Si vous cochez la case **Toujours demander des valeurs lors du test** dans la boîte de dialogue **Invite**, Framework Manager vous invite à indiquer une valeur chaque fois que vous testez l'objet. Lorsque vous mettez à jour l'objet ou que effectuez un comptage, Framework Manager utilise la valeur d'invite existante, le cas échéant.

Une invite sur un élément de requête d'un sujet de requête de modèle est uniquement associée à cet élément de requête. Une invite sur un sujet de requête de source de données est associée à la totalité du sujet de requête. Par conséquent, l'invite s'affiche lorsque vous testez n'importe quel élément de requête du sujet de requête.

## Variables SAP BW

Les variables SAP BW sont des paramètres d'une requête SAP BW configurés lors de la définition d'une requête. Lorsque vous exécutez la requête, des valeurs sont automatiquement indiquées pour les variables SAP BW. Ces valeurs se comportent comme des indicateurs d'emplacement et peuvent être traitées de différentes manières. Elles sont automatiquement exposées en tant qu'invites lors de l'exécution.

Les informations de variable SAP BW sont incluses dans une propriété personnalisée composite nommée **Variables SAP BW** qui existe uniquement lorsqu'une source de données a une ou plusieurs variables associées. La propriété **Variables SAP BW** contient une ou plusieurs propriétés composites, chacune d'entre elles devant avoir un nom unique. Chaque propriété représente une description d'une variable SAP BW. Etant donné que les informations de variable sont définies dans une propriété personnalisée, Framework Manager ne valide pas ces propriétés.

Les informations de variable SAP BW sont obtenues à l'aide de l'interface BAPI SAP BW **MDDataProviderBW::GetVariables**.

Framework Manager prend en charge ces types de variable SAP BW :

- **Caractéristique**  
Il existe deux types de variable de caractéristique : valeur de caractéristique et noeud de hiérarchie. Les variables de valeur de caractéristique sélectionnent des valeurs de caractéristique. Les variables de noeud de hiérarchie sélectionnent des valeurs à tout emplacement d'une hiérarchie de présentation.
- **Hiérarchie**  
L'utilisateur n'est pas invité à entrer de valeur car le logiciel IBM Cognos la charge automatiquement lors de l'exécution, en fonction de la hiérarchie sélectionnée. Les variables pour les hiérarchies se comportent comme des indicateurs de placement pour la hiérarchie d'une caractéristique. Toutes les valeurs des variables de hiérarchie sont en lecture seule.
- **Formule**  
L'utilisateur entre une valeur numérique lors de l'exécution. Utilisez des variables de formule s'il est nécessaire d'entrer un composant de formule uniquement lors de l'exécution d'une requête. Par exemple, vous pouvez utiliser une variable de formule pour un taux d'imposition afin de traiter le taux actuel lors de l'exécution.
- **Autorisation**  
Les variables d'autorisation sont similaires aux autres variables à la seule différence que le logiciel IBM Cognos charge automatiquement les valeurs de variable avec les données d'identification de l'utilisateur. SAP BW utilise ces données d'identification pour fournir les informations requises par une requête SAP BW à laquelle la sécurité est appliquée.

Les variables pour les hiérarchies se comportent comme des indicateurs de placement pour la hiérarchie d'une caractéristique. Toutes les valeurs des variables de hiérarchie sont en lecture seule.

### **Propriété Nom**

Cette propriété est une valeur de type chaîne.

Equivalent SAP BW : VARIABLE\_NAME

Restrictions : en lecture seule.

### **Propriété Légende**

La valeur de type chaîne de cette propriété est un élément composite dépendant de l'environnement local. Représentez chaque environnement local dans le modèle par une propriété personnalisée dont la valeur est le nom d'environnement local. Par

exemple, si les environnements locaux en-ca et fr-fr existent dans le modèle, définissez deux propriétés personnalisées nommées en-ca et fr-fr.

La valeur par défaut est obtenue à partir de SAP BW.

### Propriété Type de sélection

Les valeurs possibles sont value, interval, complex, multiValued.

Tableau 7. Equivalents de la propriété Type de sélection

Valeur	Equivalent SAP BW
value	SAP_VAR_SEL_TYPE_VALUE
interval	SAP_VAR_SEL_TYPE_INTERVAL
complex	SAP_VAR_SEL_TYPE_COMPLEX
multiValued	SAP_VAR_SEL_TYPE_COMPLEX

Restrictions : en lecture seule.

### Propriété Type d'entrée

La valeur par défaut est obtenue à partir de SAP BW.

Tableau 8. Equivalents de la propriété Type d'entrée

Valeur	Equivalent SAP BW
optional	SAP_VAR_INPUT_TYPE_OPTIONAL
mandatory	SAP_VAR_INPUT_TYPE_MANDATORY
mandatoryNotInitial	SAP_VAR_INPUT_TYPE_MANDATORY_NOT_INITIAL

Restrictions : en lecture seule.

### Propriétés Valeur basse par défaut et Valeur élevée par défaut

Chaque propriété définit une plage de valeurs.

La valeur par défaut est obtenue à partir de SAP BW.

Restrictions : La propriété **Valeur élevée par défaut** s'applique uniquement aux variables pour lesquelles la valeur **interval** est attribuée à **Type de sélection**.

### Propriété Description

Cette propriété est une valeur de type chaîne.

## Propriété Type de variable SAP BW

Les valeurs possibles sont numeric, characteristic, hierarchy ou hierarchicalNode.

La valeur par défaut est obtenue à partir de SAP BW.

Restrictions : en lecture seule.

## Propriété Type d'invite

La valeur par défaut dépend du type de variable. Si la valeur de cette propriété n'est pas une des valeurs prédéfinies, la valeur utilisée est hierarchyPickList. Les valeurs prédéfinies de la propriété Type d'invite sont les suivantes.

Tableau 9. *Equivalents de la propriété Type d'invite*

Valeur	Restrictions
typeIn	Requise pour les variables numériques et facultatif pour les valeurs de caractéristique
pickList	Facultative pour les variables de caractéristique
calendar	Uniquement pour les variables de caractéristique basées sur OCALDAY
hierarchyPickList	Facultative pour toutes les hiérarchies de présentation
notApplicable	Requise pour les variables de hiérarchie

Cette propriété permet de définir le type d'invite.

Vous pouvez améliorer les performances des invites de variables utilisant une liste de sélection ou une liste de sélection hiérarchique. Utilisez les propriétés de restriction de niveau, de nombre initial de valeurs de liste de sélection et d'utilisation de valeur par défaut afin de contrôler les performances de ces types d'invite de variable.

La modification d'une invite de liste de sélection ou de liste de sélection hiérarchique en une invite de saisie peut améliorer les performances de manière significative car ainsi, il n'est pas nécessaire que le serveur d'applications charge des valeurs dans une liste de sélection. Toutefois, vos utilisateurs doivent pouvoir entrer des valeurs de caractéristique.

Restrictions : Lecture seule pour certains types de variable, tels les caractéristiques et la formule.

## Propriété Restriction de niveau

Cette propriété est une valeur numérique.

La valeur par défaut est 1.

Cette propriété permet de réduire le nombre de valeurs de caractéristique qui chargent une liste de sélection hiérarchique. Le nombre de niveaux d'une hiérarchie à partir desquels les valeurs peuvent être obtenues est limité.

Si la valeur est égale à zéro (0), ce qui constitue la valeur par défaut, alors les valeurs de caractéristique de tous les niveaux d'une hiérarchie (si elles s'appliquent à ce type d'invite) chargent la liste de sélection. Sinon, la propriété définit une plage de niveaux séparés par des virgules à partir de laquelle des valeurs sont obtenues (le niveau d'origine est zéro).

Pour une hiérarchie irrégulière, vous devez indiquer tous les niveaux que vous souhaitez utiliser même lorsque certaines branches n'ont pas ce niveau.

Restrictions : S'applique uniquement aux variables de noeud hiérarchiques ayant la valeur **hierarchyPickList** pour **Type d'invite**.

### Propriété d'utilisation des valeurs par défaut

Cette propriété est une propriété booléenne qui détermine quand les valeurs par défaut sont utilisées. Si cette propriété à la valeur **true**, les utilisateurs ne sont pas invités à entrer la variable associée et la valeur par défaut est toujours appliquée.

Cette propriété permet d'attribuer une valeur unique à la variable. Les utilisateurs ne sont pas invités à entrer la valeur d'une variable et par conséquent, le serveur IBM Cognos BI ne charge pas de valeur dans la liste de sélection. Toutefois, les utilisateurs ne peuvent plus changer la valeur d'une variable.

La valeur par défaut est **false**.

### Propriété Afficher la clé et la légende

Pour afficher les clés et les légendes de variables SAP BW, attribuez la valeur **true** à cette propriété. Cette dernière s'applique uniquement aux invites de noeud hiérarchique et de liste de sélection.

La valeur par défaut est **false**.

### Propriété Nombre initial de valeurs de liste de sélection

Cette propriété définit le nombre initial de valeurs permettant de charger une liste de sélection, une liste de sélection hiérarchique ou une invite.

La valeur par défaut est zéro (0), ce qui correspond à tous les éléments.

## Valeurs de propriété de variable numérique (SAP BW)

Les propriétés de variables suivantes s'appliquent à des variables numériques :

Propriété	Valeur par défaut
Type	numérique
Légende	
Type de sélection	valeur
Type d'entrée	obtenue à partir de SAP BW
Valeur basse par défaut	

Propriété	Valeur par défaut
Valeur élevée par défaut	
Type d'invite	typeIn
Utiliser la valeur par défaut	false

Vous pouvez modifier les valeurs par défaut d'une variable numérique sauf pour la propriété **Type d'invite**, qui est accessible en lecture seule.

## Valeurs de propriété de variable de caractéristique (SAP BW)

Il existe deux types de variable de caractéristique : valeur de caractéristique et noeud hiérarchique. Les variables de valeur de caractéristique sélectionnent les valeurs de caractéristique. Les variables de noeud hiérarchique sélectionnent des valeurs à partir de n'importe quelle position dans une hiérarchie de présentation.

### Valeurs de propriété de variable de valeur de caractéristique

Les propriétés de variable suivantes s'appliquent à des variables de valeur de caractéristique :

Propriété	Valeur par défaut
Type	caractéristique
Légende	
Type de sélection	obtenue à partir de SAP BW
Type d'entrée	obtenue à partir de SAP BW
Valeur basse par défaut	Si le type d'entrée est une valeur ou complexe, la propriété par défaut est affichée.  Si le type d'entrée est un intervalle, la propriété basse par défaut est affichée. Cette valeur est obtenue à partir de SAP BW.
Valeur élevée par défaut	Si le type d'entrée est une valeur ou complexe, la propriété par défaut est affichée.  Si le type d'entrée est un intervalle, la propriété élevée par défaut est affichée. Cette valeur est obtenue à partir de SAP BW.
Type d'invite	typeIn ou pickList  Dépend du nombre de membres dans la dimension référencée.  Si la valeur est incorrecte, typeIn est utilisé.
Utiliser la valeur par défaut	false
Afficher la clé et la légende	false
Nombre initial de valeurs de liste de sélection	zéro (0)

Une variable de valeur de caractéristique pour la dimension 0CALDAY est affichée dans le modèle sous la forme d'une date. La propriété **Type de données** a pour valeur **xsdDate** et la propriété **Type d'invite** a pour valeur **calendrier**. La propriété **Type d'invite** est accessible en lecture seule pour la dimension 0CALDAY.



## Valeurs de propriété de variable de noeud hiérarchique

Les propriétés de variable suivantes s'appliquent à des variables de noeud hiérarchique :

Propriété	Valeur par défaut
Type	caractéristique
Légende	
Type de sélection	obtenue à partir de SAP BW
Type d'entrée	obtenue à partir de SAP BW
valeur basse par défaut	
Valeur élevée par défaut	
Type d'invite	Liste de sélection de hiérarchie  Vous pouvez remplacer la valeur de la propriété <b>Type d'invite</b> par <b>typeIn</b> ou <b>pickList</b> .
Restriction de niveau	zéro (0)
Utiliser la valeur par défaut	false

## Invites de sélection (SAP BW)

Chaque invite de sélection contient un nombre prédéfini de valeurs. Ces valeurs sont déterminées par la propriété **Nombre maximal de valeurs**.

Si le nombre de valeurs réel est inférieur ou égal au nombre de valeurs par défaut, l'invite est générée en tant qu'invite de sélection unique. Si le nombre de valeurs réel dépasse le nombre par défaut, deux invites sont générées dans cet ordre :

- Un paramètre de plage limite avec une valeur de départ de **1** et une valeur de fin déterminée par la propriété **Nombre maximal de valeurs**

Ce paramètre est de type `xsdUnsignedLong` et facultatif. Le nom du paramètre est le nom de l'invite d'origine suivi par `_range_prompt`. La légende dépend des paramètres régionaux. S'il s'agit d'un modèle multilingue, vous devez stocker le modèle pour la légende dans un fichier de messages.

- Une invite de sélection contenant le nombre de valeurs par défaut

---

## Ajout de règles métier (SAP BW)

Les règles métier qui ont été créées dans SAP BW sont importées dans IBM Cognos Framework Manager. Vous pouvez ajouter des règles métier supplémentaires à votre modèle pour affiner les données extraites et mettre les informations appropriées à disposition des utilisateurs.

Créer des règles métier et les stocker dans le modèle plutôt que dans des rapports présentent de nombreux avantages. Vous gagnez du temps car il n'est plus nécessaire de recréer les règles métier chaque fois qu'elles sont requises. Les règles métier assurent une cohérence car vos utilisateurs utilisent tous les mêmes définitions. Par exemple, la signification de l'expression Marges inférieures est identique dans toute l'organisation. Elles sont faciles à mettre à jour car vous les gérez de manière centralisée. Ainsi, tous les rapports sont mis à jour automatiquement à mesure que les règles évoluent. Par exemple, si la définition de

l'expression Marge inférieure change, tous les rapports qui utilisent le calcul Marge inférieure sont mis à jour automatiquement. Les règles métier permettent d'améliorer la sécurité.

### Procédure

1. Ajoutez des calculs de sorte que vos utilisateurs puissent inclure des données calculées dans leurs rapports («Création d'un calcul (SAP BW)»).
2. Créez et appliquez des filtres permettant de limiter les données extraites par un sujet de requête («Création d'un filtre (SAP BW)», à la page 242).
3. Ajoutez des invites qui s'affichent automatiquement chaque fois qu'une dimension ou un sujet de requête est utilisé dans un rapport. Les consommateurs du rapport sont alors invités à filtrer les données («Définition d'un contrôle d'invite (SAP BW)», à la page 228).
4. Utilisez des paramètres de session («Création d'un paramètre de session (SAP BW)», à la page 248) et des tables de mappage des paramètres («Création d'une table de mappage des paramètres (SAP BW)», à la page 246) pour résoudre les expressions de façon dynamique.
5. Créez un filtre de sécurité pour contrôler les données affichées pour vos utilisateurs lorsqu'ils configurent leurs rapports («Ajout de sécurité des données», à la page 261).

## Création d'un calcul (SAP BW)

Les caractères de ponctuation, tels que le point d'interrogation (?), doivent être exprimés en code de caractère ASCII 7 bits. Si vous tapez un caractère de ponctuation à l'aide d'un clavier multi-octets activé, prenez soin d'entrer la représentation ASCII 7 bits du caractère. Par exemple, tapez Alt+063 pour le point d'interrogation.

Évitez de nommer le calcul à l'aide de caractères utilisés pour les opérateurs d'expression. Des erreurs de syntaxe sont susceptibles de se produire lorsque l'expression est évaluée. Par exemple, un calcul nommé `Margin * 10` génère des erreurs lorsqu'il est utilisé dans une expression telle que `[Margin * 10] < 20`.

Dans les expressions, un type dimensionnel spécifique peut être requis pour les opérandes d'un opérateur ou d'une fonction. Lorsqu'un opérande n'est pas du type requis, une ou plusieurs règles de conversion peuvent être appliquées pour convertir l'opérande au type approprié. Étant donné que les règles de conversion ne sont pas appliquées aux expressions des sujets de requête de modèle, vérifiez que ces expressions sont valides sans vous baser sur les règles de conversion. Pour plus d'informations sur les règles de conversion, voir le document *IBM Cognos Report Studio User Guide*.

Si vous insérez dans le calcul une fonction définie par l'utilisateur importée, assurez-vous que le nom de la fonction ne reprend pas le nom d'un fournisseur. Par exemple, si le nom de la fonction définie par l'utilisateur est `CHAR`, une erreur est générée lorsque vous testez la fonction via la boîte de dialogue **Définition du calcul** car ce nom est considéré comme étant identique à `char` dans Microsoft SQL Server. Pour plus d'informations sur les noms de fonction utilisés dans votre base de données, voir la documentation produit de la base de données.

Lors de la requête, IBM Cognos Framework Manager renvoie une valeur nulle pour les calculs qui contiennent un diviseur dont la valeur est égale à zéro.


Framework Manager ne parvient pas à détecter les erreurs de division par zéro dans des fonctions telles que average et mod, car l'opérateur de division n'est pas explicite.

Vous pouvez appliquer un calcul autonome à un ou plusieurs sujets de requête ou dimensions afin de fournir des données calculées à un rapport, ou inclure ce calcul dans un pack afin de le rendre disponible pour vos utilisateurs. Le fait de déplacer un calcul autonome ou un raccourci vers celui-ci dans un dossier vous permet d'organiser les objets du modèle de façon plus efficace.

## Procédure de création d'un calcul

### Procédure

1. Cliquez sur l'espace-noms ou le dossier et, depuis le menu **Actions**, cliquez sur **Créer, Calcul**.
2. Dans la zone **Nom**, saisissez un nom à attribuer au calcul.
3. Définissez l'expression.

Objectif	Action
Ajouter des éléments	Dans l'onglet <b>Modèle</b> , cliquez sur un élément de requête, un filtre, ou un calcul, puis cliquez sur la flèche.
Ajouter des fonctions	Dans l'onglet <b>Fonctions</b> , sélectionnez un composant et cliquez sur la flèche.
Ajouter des paramètres	Dans l'onglet <b>Paramètres</b> , sélectionnez un paramètre et cliquez sur la flèche.
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cliquez sur le bouton d'options, cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes à renvoyer.</p>  <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions et de sujets de requête.</p>
Redéfinir des paramètres de session	Cliquez sur le bouton d'options, puis sur <b>Définir</b> , entrez une valeur dans la zone <b>Remplacer la valeur</b> et cliquez sur <b>OK</b> .
Redéfinir des valeurs d'invite	<p>Cliquez sur le bouton d'options, puis sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>

4. Pour tester le calcul, cliquez sur le bouton de **test** .

Vous pouvez uniquement tester des calculs qui contiennent des éléments de requête. Si un calcul contient une fonction, par exemple, `_add_days`, le bouton **Tester l'exemple** n'est pas disponible.

5. Cliquez sur **OK**.
6. Modifiez la propriété **Type de données** pour identifier le type de donnée renvoyé par le calcul.  
Les studios IBM Cognos utilisent ces informations pour formater les données renvoyées par le calcul.

## Résultats

Pour en savoir davantage sur les fonctions, reportez-vous à la section Annexe C, «Utilisation de l'éditeur d'expression», à la page 403.

## Création d'un filtre (SAP BW)

Par exemple, vous pouvez utiliser la fonction `in_range` pour créer un filtre qui extrait les données pour les produits introduits dans un délai imparti déterminé. La syntaxe de cet exemple se présente comme suit :

```
[gosales_goretailers].[Products].[Introduction  
date]  
in_range {Feb 14, 1999 : July 14, 2007}
```

**Remarque :** lorsque vous utilisez une fonction de date ou d'heure, vous devez utiliser une horloge au format 24 heures. IBM Cognos Framework Manager ne prend pas en charge les mentions "a.m." et "p.m." dans les expressions. Par exemple, vous utiliserez 20:00 pour indiquer 8 p.m.

Vous pouvez limiter les données représentées par les dimensions ou les sujets de requête dans un projet en créant un filtre de sécurité. Le filtre de sécurité permet de contrôler les données visibles par vos utilisateurs lorsqu'ils configurent leurs rapports.

Vous pouvez également appliquer des restrictions afin de limiter les données extraites par les requêtes d'un pack.

Framework Manager prend en charge les filtres autonomes et les filtres imbriqués.

- Utilisez un filtre autonome lorsque vous souhaitez réutiliser l'expression.  
Vous pouvez ajouter un filtre autonome à un ou plusieurs sujets de requête ou dimensions afin de limiter les données extraites par la requête lorsque le sujet de requête ou la dimension filtrés sont utilisés dans un rapport, ou vous pouvez inclure ce filtre autonome dans un pack afin de le rendre disponible pour vos utilisateurs. Le fait déplacer un filtre autonome ou un raccourci vers celui-ci dans un dossier vous permet d'organiser les objets du modèle de façon plus efficace.
- Utilisez un filtre imbriqué lorsque vous souhaitez utiliser un filtre avec une seule dimension ou un seul sujet de requête.  
Vous pouvez créer un filtre imbriqué lors de la modification d'une dimension, d'un sujet de requête de source de données relationnelle ou d'un sujet de requête de modèle.  
Si vous démarrez à l'aide d'un filtre imbriqué, vous pouvez ensuite le convertir en une expression autonome que vous pouvez appliquer à d'autres dimensions

ou sujets de requête. **Astuce** : cliquez avec le bouton droit de la souris sur l'expression de filtre sous l'onglet **Filtres**, puis cliquez sur **Convertir en filtre autonome**.

Lorsque vous imbriquez un filtre, le sujet de requête de source de données doit avoir une relation avec n'importe quel sujet de requête référencé par l'expression. Cette relation est nécessaire même si l'expression fait référence à un sujet de requête de modèle basé sur la même table que le sujet de requête de source de données dans lequel vous incorporez l'expression.

Pour créer un filtre sur un sujet de requête non associé, procédez de l'une des façons suivantes :

- Vérifiez qu'il existe un chemin de jointure entre le nouveau sujet de requête nouvelle et celui contenant le filtre.
- Fondez le filtre imbriqué sur un élément de requête basé sur le sujet de requête de source de données souhaité.
- Convertissez le filtre en un filtre autonome, de sorte qu'il ne fasse pas partie du sujet de requête.
- Créez un filtre autonome qui fait référence à l'objet imbriqué.

Vous pouvez être amené à consulter les rubriques associées suivantes :

- Filtres de sécurité «Ajout de sécurité des données», à la page 261
- Fonctions Annexe C, «Utilisation de l'éditeur d'expression», à la page 403
- Paramètres «Création d'une table de mappage des paramètres (SAP BW)», à la page 246
- Paramètres de session «Création d'un paramètre de session (SAP BW)», à la page 248

Vous pouvez également appliquer des restrictions afin de limiter les données extraites par les requêtes d'un pack.


## Procédure de création de filtres

### Procédure

1. Effectuez l'une des opérations suivantes :
  - Pour créer un filtre autonome, cliquez sur l'espace-noms ou le dossier, puis sur **Actions, Créer, Filtre**.
  - Si vous souhaitez créer un filtre intégré, cliquez deux fois sur la dimension ou le sujet de requête et cliquez sur l'onglet **Filtres**, puis sur **Ajouter**.
2. Dans la zone **Nom**, saisissez un nom à attribuer au filtre.
3. Définissez l'expression.
 

**Astuce** : Si le filtre comprend une expression non valide, consultez la zone **Conseils** dans l'éditeur d'expression pour plus d'informations.

Objectif	Action
Ajouter des éléments de requête et des filtres	Dans l'onglet <b>Modèle</b> , faites glisser les objets souhaités vers la zone <b>Définition de l'expression</b> .
Ajouter des fonctions	Dans l'onglet <b>Fonctions</b> , faites glisser les fonctions vers la zone <b>Définition de l'expression</b> .

Objectif	Action
Ajouter des paramètres	Dans l'onglet <b>Paramètres</b> , faites glisser les paramètres vers la zone <b>Définition de l'expression</b> .
Extraire toutes les données et afficher un nombre spécifique de lignes	<p>Cliquez sur le bouton d'options, cochez la case <b>Restreindre le nombre maximal de lignes à renvoyer</b> et entrez le nombre requis de lignes à renvoyer.</p>  <p>Ce paramètre ne permet pas d'améliorer les performances pour l'extraction des données lors du test de dimensions, de sujets de requête et d'ensembles de requêtes.</p>
Redéfinir des paramètres de session	Cliquez sur le bouton d'options, puis sur <b>Définir</b> , entrez une valeur dans la zone <b>Remplacer la valeur</b> et cliquez sur <b>OK</b> .
Redéfinir des valeurs d'invite	<p>Cliquez sur le bouton d'options, puis sur <b>Invites</b>.</p> <p>La boîte de dialogue <b>Gestionnaire d'invites de modèle</b> s'affiche avec toutes les invites et les valeurs correspondantes figurant dans le modèle.</p>

4. Cliquez sur **OK**.

## Résultats

Vous pouvez également appliquer des restrictions afin de limiter les données extraites par les requêtes d'un pack. Pour plus d'informations, voir «Restrictions», à la page 309.

Vous pouvez être amené à consulter les rubriques associées suivantes :

- «Ajout de sécurité des données», à la page 261
- Annexe C, «Utilisation de l'éditeur d'expression», à la page 403
- «Création de tables de mappage des paramètres», à la page 160
- «Création de paramètres de session», à la page 163
- «Test de sujets de requête ou d'ensembles de requêtes», à la page 93
- «Paramètres et invites obligatoires», à la page 169

## Application d'un filtre (SAP BW)

Vous pouvez incorporer un filtre autonome dans les dimensions ou les sujets de requête, mais si vous souhaitez une utilisation différente pour chaque filtre incorporé, vous devez créer différentes versions du filtre autonome. Sinon, il se peut que vos utilisateurs soient obligés de répondre à une invite que vous pensiez être facultative s'il existe une instance dans laquelle l'utilisation a été définie

comme obligatoire. Pour plus d'informations sur les invites facultatives et obligatoires, voir «Paramètres et invites obligatoires», à la page 169.

Par exemple, dans le sujet de requête A, vous incorporez un filtre autonome que vous définissez comme étant facultatif. Dans le sujet de requête B, vous le définissez comme étant obligatoire. Lorsque vos utilisateurs créent un rapport qui utilise les deux sujets de requête, il leur est demandé de choisir des valeurs dans les deux filtres, même dans celui qui est défini comme étant facultatif. Toutes les instances du filtre sont considérées comme étant obligatoires lorsqu'elles sont utilisées dans la même requête. La solution consiste à créer différentes versions du filtre, chacun ayant un nom distinct.

## Procédure d'application d'un filtre

### Procédure

1. Créez un filtre.
2. Sélectionnez le filtre, puis cliquez sur **Actions, Editer la définition**.
3. Cliquez sur l'onglet **Filtres**, puis faites glisser le filtre que vous avez créé vers la zone **Filtres**.
4. Sélectionnez une valeur d'utilisation pour votre filtre.

Valeur d'utilisation	Description
Toujours	Utilisez cette valeur d'utilisation pour que les données spécifiées soient filtrées dans les rapports. Par exemple, votre entreprise peut stocker des informations obsolètes sans vouloir générer des rapports sur celles-ci.  <b>Toujours</b> est la valeur d'utilisation par défaut.
Mode de conception uniquement	Extrait un petit sous-ensemble des données pour l'exemple de rapport. Utilisez cette valeur d'utilisation lorsque vous n'avez pas besoin de voir toutes les données, par exemple, lorsque vous testez un sujet de requête.  Pour appliquer des filtres de mode de conception dans Framework Manager, sélectionnez l'option <b>Appliquer tous les filtres de mode de conception appropriés lors du test</b> . Cette option est disponible dans l'onglet <b>Tester les paramètres</b> .  Les utilisateurs peuvent avoir besoin du filtrage de mode de conception dans Query Studio quand ils veulent se concentrer sur la conception de la présentation et du format d'un rapport, et ne pas extraire toutes les données quand ils travaillent. Pour accéder au filtrage de mode de conception dans Query Studio, exécutez le rapport avec un sous-ensemble de données.

Valeur d'utilisation	Description
Facultatif	<p>Indique si le filtre est facultatif. L'utilisateur est invité à filtrer les données et peut laisser la zone vide. Si l'invite est vide, Framework Manager ignore le filtre et extrait toutes les données de la dimension ou du sujet de requête.</p> <p>La syntaxe ?? est requise pour les invites facultatives.</p> <p>Faites appel à cette valeur d'utilisation si les utilisateurs veulent contrôler à quel moment appliquer le filtre. Par exemple, vous souhaitez afficher parfois un pays et d'autres fois visualiser les données pour tous les pays. Un filtre facultatif pour les pays se présente comme suit :</p> <pre> ([GeoNamespace].[Countries] .[CountryName] = ?WhichCountry?) </pre>

- Si vous souhaitez afficher le code SQL, cliquez sur l'onglet **Informations sur la requête**.
- Cliquez sur **OK**.

## Création d'une table de mappage des paramètres (SAP BW)

Les tables de mappage des paramètres sont semblables aux tables de recherche de source de données. Chaque table de mappage des paramètres comporte deux colonnes, l'une contenant la clé et l'autre contenant la valeur représentée par la clé. Vous pouvez entrer manuellement les clés et les valeurs, les importer à partir d'un fichier ou les baser sur des éléments de requête du modèle.

Vous pouvez également exporter des tables de mappage des paramètres dans un fichier. Pour modifier la table de mappage des paramètres, vous pouvez exporter les valeurs qu'elle contient dans un fichier, effectuer des ajouts ou des modifications, puis l'importer à nouveau dans IBM Cognos Framework Manager. Cela s'avère particulièrement utile lorsque vous utilisez des tables de mappage des paramètres volumineuses et complexes.

Toutes les clés de table de mappage des paramètres doivent être uniques pour que Framework Manager puisse extraire la valeur correcte de façon cohérente. Vous ne devez pas placer la valeur de paramètre entre guillemets. Vous pouvez utiliser des guillemets dans l'expression dans laquelle le paramètre est utilisé.

La valeur d'un paramètre peut être un autre paramètre. Toutefois, vous devez placer la totalité de la valeur entre signes dièses (#). L'imbrication de paramètres en tant que valeurs est limitée à cinq niveaux.

Lorsque vous utilisez une table de mappage des paramètres en tant qu'argument sur une fonction, vous devez utiliser le symbole du pourcentage (%) au lieu du symbole du dollar (\$).



Attribuez un alias à un élément de requête dont le nom se compose en partie d'une table de mappage des paramètres dans son nom et pour ajouter des noms multilingues à l'objet dans l'onglet **Langue** (sous-fenêtre **Propriétés**).

**Remarque :** si vous employez des métadonnées SAP BW, vous ne pouvez pas utiliser un élément de requête pour générer les clés et les valeurs d'une table de mappage des paramètres.

## Procédure de création manuelle d'une table de mappage des paramètres

### Procédure

1. Cliquez sur le dossier **Tables de mappage de paramètres**, puis sur **Actions, Créer, Table de mappage de paramètres**.
2. Dans la zone **Nom**, saisissez un nom à attribuer à la nouvelle table de mappage des paramètres.
3. Cliquez sur **Entrez manuellement les clés de paramètre et/ou importez-les à partir d'un fichier**, puis sur **Suivant**.
4. Effectuez l'une des opérations suivantes :
  - Pour entrer manuellement des valeurs, cliquez sur **Nouvelle clé**, entrez une clé et appuyez sur la touche de tabulation pour saisir une valeur pour cette clé.
  - Pour importer des clés et des valeurs, cliquez sur **Importer un fichier** et identifiez l'emplacement du fichier .csv ou .txt approprié. Pour un fichier .txt à utiliser pour l'importation, les valeurs doivent être séparées par des tabulations et le fichier doit être enregistré au format UTF8 ou Unicode. Les fichiers texte ANSI ne sont pas pris en charge.

**Remarque :** Si vous prévoyez d'utiliser un paramètre dans un sujet de requête de source de données, la valeur doit utiliser la ponctuation anglaise. Cela signifie que vous devez utiliser un point (.) pour représenter une décimale et une virgule (,) pour séparer des valeurs dans les listes.

5. Modifiez les paramètres existants selon les besoins.

Objectif	Action
Affecter une valeur par défaut	<p>Dans la zone <b>Valeur par défaut</b>, saisissez une valeur.</p> <p>Si la clé utilisée dans une expression n'est pas mappée, la valeur par défaut est utilisée.</p> <p>La définition d'une valeur par défaut est facultative. Cependant, si aucune valeur par défaut n'est fournie, une clé non mappée risque de générer une erreur.</p>

Objectif	Action
Charger à la demande	Le chargement d'une table de mappage des paramètres contenant un grand nombre de clés peut être long lors de l'exécution d'un rapport. Vous pouvez limiter le chargement aux clés qui sont requises pour un rapport spécifique en définissant <b>Charger à la demande</b> sur true.  Par défaut, la valeur est définie sur false et toutes les clés sont chargées.
Supprimer un paramètre	Sélectionnez une ligne et cliquez sur <b>Supprimer</b> .
Modifier un paramètre	Sélectionnez la ligne à modifier, cliquez sur le bouton <b>Editer</b> et entrez une valeur.
Effacer toutes les clés et les valeurs	Cliquez sur <b>Clear Map</b> (Effacer la mappe).

6. Cliquez sur **Terminer**.

### Valeurs par défaut et tables de mappage des paramètres

Les propriétés **Valeur basse par défaut** et **Valeur élevée par défaut** peuvent contenir des expressions qui utilisent des tables de mappage des paramètres. Vous pouvez utiliser des tables de mappage des paramètres pour définir une valeur pour une variable de devise cible basée sur les paramètres régionaux de l'utilisateur. Par exemple, vous définissez une table de mappage des paramètres qui fournit un code de devise ISO. La valeur pour la propriété **Valeur basse par défaut** peut être définie en tant que

```
#$Currency_Map[runLocale]#
```

Cette table de mappage des paramètres est utilisée lorsque la devise cible variable SAP BW est utilisée dans un rapport.

Il s'agit des seules les propriétés relatives aux variables SAP BW pouvant utiliser des tables de mappage des paramètres.

## Création d'un paramètre de session (SAP BW)

Par exemple, l'ID utilisateur et la langue préférée sont deux paramètres de session. Les paramètres de session étant des paires clé-valeur, vous pouvez envisager chaque paramètre de session comme une entrée d'une table de mappage des paramètres appelée Paramètres de session. Vous pouvez utiliser un paramètre de session de la même façon que vous utilisez une entrée de table de mappage des paramètres, même si la syntaxe pour les paramètres de session est légèrement différente.

Il existe deux types de paramètre de session : environnement et modèle.

les paramètres de session d'environnement sont prédéfinis et stockés dans Content Manager. Par défaut, les paramètres de session suivants s'affichent dans Framework Manager :

- **runLocale**

Renvoie le code de la langue active en cours dans Framework Manager. Le contenu de modèle est affiché dans cette langue.

Vous pouvez modifier la langue active à tout moment pour votre session en cours uniquement. Dans les sessions futures, le modèle continuera de s'ouvrir dans la langue utilisée lors de la conception. Pour plus d'informations, voir la section «Ajout d'une langue à un projet», à la page 129.

- **account.defaultName**

Indique le nom de l'utilisateur en cours tel qu'il est défini dans le fournisseur d'authentification. Par exemple, le prénom et le nom de l'utilisateur.

Si vous vous connectez en tant qu'anonyme, **Anonyme** est affiché.

- **account.personalInfo.userName**

Indique l'ID utilisateur sous lequel la connexion à IBM Cognos BI est établie.

Si vous vous connectez en tant qu'anonyme, ce paramètre n'est pas affiché.

- **current\_timestamp**

Indique la date et l'heure en cours.

- **machine**

Indique le nom de l'ordinateur sur lequel Framework Manager est installé.

Si votre source d'authentification prend en charge d'autres paramètres et que vous avez entré des informations les concernant dans la source d'authentification, d'autres paramètres de session sont affichés, tels que `account.personalInfo.email` ou `account.personalInfo.surname`.

Vous pouvez définir d'autres paramètres en utilisant les paramètres de session de modèle. Les paramètres de session de modèle sont stockés dans une table de mappage des paramètres appelée `_env`. Ils sont définis dans le projet et peuvent être publiés avec un pack.

Les valeurs définies pour les paramètres de session de modèle doivent être comprises dans la portée des objets du modèle Framework Manager. Cette portée peut inclure l'utilisation des paramètres de session d'environnement existants, ainsi que des valeurs statiques.

Chaque paramètre de session doit porter un nom et posséder une valeur par défaut. Vous pouvez définir une valeur de substitution pour tester les résultats renvoyés par cette valeur. La valeur de substitution est valide uniquement lorsque le modèle est ouvert ; elle n'est pas enregistrée lors de la sauvegarde du modèle. En l'absence de valeur de substitution, Framework Manager utilise la valeur par défaut lorsqu'il exécute une requête qui contient un paramètre de session.

Les règles qui régissent l'utilisation de paramètres sont les suivantes :

- Toutes les valeurs de retour possibles doivent posséder le même type de données.
- Une seule valeur peut être définie.

## Procédure de création de paramètres de session

### Procédure

1. Cliquez sur **Projet, Paramètres de session**.
2. Cliquez sur **Nouvelle clé** et entrez une clé et une valeur de paramètre de session.
3. Sélectionnez le mode de traitement de la valeur de remplacement.

- Afin d'éviter d'avoir à définir la valeur de remplacement chaque fois que vous éditez le projet, définissez le paramètre de session en tant que valeur.
- Afin d'éviter d'avoir à supprimer le paramètre de projet chaque fois que vous le publiez, définissez le paramètre de session en tant que remplacement de session.

4. Modifiez les paramètres existants selon les besoins.

Objectif	Action
Modifier la valeur de paramètre	Cliquez sur la ligne qui contient la valeur à modifier, puis sur <b>Editer</b> et entrez une valeur.
Affecter une valeur par défaut	Dans la zone <b>Valeur par défaut</b> , saisissez une valeur. Framework Manager utilise la valeur par défaut si une clé a une valeur non valide.
Supprimer un paramètre	Cliquez sur une ligne, puis sur le bouton <b>Supprimer</b> . Vous ne pouvez pas supprimer un paramètre de session d'environnement.
Effacer une valeur de remplacement	Cliquez sur une ligne, puis sur <b>Effacer le remplacement</b> .

5. Cliquez sur **OK**.

---

## Organisation du modèle (SAP BW)

Une meilleure pratique consiste à créer plusieurs vues ou couches dans le modèle :

- Conservez les métadonnées de la source de données dans un espace-noms ou dossier distinct.

Dans IBM Cognos Framework Manager, il s'agit de la vue d'importation.

- Créez au moins un espace-noms ou dossier facultatif pour résoudre les difficultés relatives à l'interrogation à l'aide de sujets de requête ou d'objets dimensionnels.

Pour utiliser IBM Cognos Analysis Studio ou n'importe quelle requête OLAP, il doit y avoir un espace-noms ou un dossier dans le modèle qui représente les métadonnées avec objets dimensionnels.

- Créez au moins un espace-noms ou dossier pour la vue métier globale des métadonnées contenant des raccourcis vers les dimensions ou sujets de requête.

Dans Framework Manager, il s'agit de la vue métier. Utilisez des concepts métier pour modéliser la vue métier. Un même modèle peut contenir de nombreuses vues métier, chacune adaptée à un groupe d'utilisateurs différent. Publiez les vues métier.

La sécurité peut être définie dans l'une des vues. Cela dépend de vos besoins métier. Par exemple, si devez empêcher tous les utilisateurs de visualiser un objet, vous ajoutez de la sécurité à ce dernier dans la vue d'importation. En règle générale, la sécurité est appliquée dans la vue métier.

Pour organiser un modèle, vous pouvez inclure des métadonnées dans plusieurs dossiers en utilisant des raccourcis ou vous pouvez créer des espaces-noms ou des dossiers.

## Raccourcis (SAP BW)

Par exemple, vous créez des dossiers nommés Commandes, Produits et Clients. Si vous souhaitez que les dossiers Commandes et Clients contiennent la même dimension, vous devez créer un raccourci vers la dimension et l'ajouter aux deux dossiers.

**Remarque :** Deux raccourcis vers des espaces-noms ou des dossiers ne doivent pas avoir le même nom dans un modèle. Pour les autres types de raccourci (par exemple, un raccourci de sujet de requête), le nom doit être unique au sein de l'espace-noms parent.

Lorsque vous créez un raccourci, IBM Cognos Framework Manager ne définit pas les propriétés **Infobulle** et **Description**. Si vous ne définissez pas ces propriétés, les valeurs affichées dans les studios IBM Cognos sont les mêmes que celles définies dans l'objet auquel le raccourci fait référence.

**Astuce :** Pour accéder à l'objet auquel le raccourci fait référence, cliquez avec le bouton droit de la souris sur le raccourci et cliquez sur **Accéder à la cible**.

Les raccourcis sont moins souples que les objets de modèle du point de vue de la présentation, mais leur maintenance est beaucoup plus légère car ils sont automatiquement mis à jour dès que l'objet cible est mis à jour. Si la maintenance est l'une de vos principales préoccupations et que vous n'avez pas besoin de personnaliser l'apparence du sujet de requête, utilisez des raccourcis.

IBM Cognos Framework Manager dispose de deux types de raccourci :

- les raccourcis ordinaires, qui sont une simple référence à l'objet cible ;
- les raccourcis d'alias, qui se comportent comme s'ils étaient une copie de l'objet d'origine avec un comportement complètement indépendant. Les raccourcis d'alias ne sont disponibles que pour les sujets de requête et les dimensions.

### Raccourcis et dimensions

Les raccourcis permettent de gérer moins de dimensions. Vous pouvez conserver des dimensions dans la vue d'importation et conserver des raccourcis dans la vue métier.

Lorsque vous créez un raccourci vers une dimension, vous ne pouvez pas personnaliser les éléments de requête qui se trouvent dans le raccourci. La totalité de la dimension est incluse dans le raccourci.

La sécurité que vous spécifiez pour un objet est transmise aux raccourcis qui font référence à l'objet sécurisé. Si vous disposez d'un raccourci vers un objet sécurisé, seuls les utilisateurs autorisés à afficher l'objet sécurisé peuvent voir le raccourci dans le pack publié.

Cliquez avec le bouton droit de la souris sur les sujets de requête, les dimensions ou les dossiers vers lesquels vous souhaitez créer des raccourcis et cliquez sur **Créer, Raccourci**.

## Création de dossiers ou d'espaces-noms (SAP BW)

Vous pouvez créer des dossiers ou des espaces-noms afin d'organiser les objets dans le modèle.

L'information la plus importante à connaître à propos des espaces-noms est qu'une fois la création du rapport commencée, tout changement de nom d'un espace-noms publié aura une incidence sur votre contenu IBM Cognos. En effet, le changement du nom de l'espace-noms modifie les identificateurs des objets publiés dans les métadonnées. Etant donné que l'espace-noms fait partie de l'identificateur d'objet dans IBM Cognos Framework Manager, chaque espace-noms doit avoir un nom unique dans le modèle. Chaque objet de l'espace-noms doit également avoir un nom unique. Une partie de la stratégie d'élaboration des groupes sous forme de schémas en étoile est de placer les raccourcis dans un espace-noms distinct, ce qui crée automatiquement un identificateur unique pour chaque objet dans l'espace-noms. Pour les bases de données relationnelles, cela vous permet d'utiliser le même nom pour les raccourcis vers des dimensions conformes dans différents groupes sous forme de schémas en étoile.

Lors de la prochaine tentative d'exécution d'une requête, d'un rapport ou d'une analyse sur le modèle mis à jour, vous obtenez une erreur. Si vous devez renommer l'espace-noms que vous avez publié, utilisez la commande **Analyze Publish Impact** pour identifier les rapports sur lesquels cela aura une incidence.

Les dossiers sont beaucoup moins compliqués que les espaces-noms. Ils n'ont qu'une fonction purement organisationnelle et n'ont aucune incidence sur l'identificateur des objets ou sur votre contenu. Vous pouvez créer des dossiers pour organiser les objets par sujet ou domaine fonctionnel. Cela facilite l'accès aux métadonnées, en particulier dans les projets volumineux.

Le principal inconvénient des dossiers réside dans le fait qu'ils nécessitent des noms uniques pour tous les sujets de requête, dimensions et raccourcis. Ils ne représentent donc pas la solution idéale pour contenir des objets partagés.

**Astuce :** Lorsque vous affichez des métadonnées sous l'onglet **Diagramme**, vous pouvez développer ou réduire les dossiers et les espaces-noms. A partir du menu **Diagramme**, cliquez sur **Tout réduire** ou **Tout développer**.

Si vous définissez la sécurité sur un dossier dans lequel vous placez ensuite des objets, vérifiez que les exclusions ont été correctement définies.

Pour les métadonnées SAP BW, les raccourcis vers les espaces-noms ne sont pas pris en charge.

## Procédure de création d'un dossier

### Procédure

1. Cliquez sur **Actions, Créer, Dossier**.
2. Dans la zone **Nom du dossier**, attribuez un nom au nouveau dossier.
3. Cliquez sur **Suivant**.
4. Indiquez si vous souhaitez déplacer les objets ou pour créer des raccourcis :
  - Pour déplacer les objets sélectionnés dans le dossier, cliquez sur **Déplacer les éléments sélectionnés vers le nouveau dossier**. Lorsque vous déplacez un objet qui participe à une relation, la relation à cet objet est également déplacée.
  - Pour créer des raccourcis faisant référence aux objets sélectionnés, cliquez sur **Créer un raccourci pour les objets sélectionnés**. Ne sélectionnez pas tous les objets de l'espace-noms pour éviter de créer une structure récursive dans le pack publié.

5. Sélectionner les données à ajouter au dossier.
6. Cliquez sur **Terminer**.

## **Procédure de création d'un espace-noms**

### **Procédure**

1. Cliquez sur **Actions, Créer, Espace-noms**.
2. Cliquez avec le bouton droit de la souris sur l'espace-noms, cliquez sur **Renommer** et attribuez à l'espace-noms un nom descriptif unique.
3. Ajoutez des objets en important des métadonnées, ou en déplaçant vers l'espace-noms des objets de modèle ou des raccourcis vers des objets.





---

## Chapitre 7. Publication de packs

Vous pouvez publier un pack pour mettre les métadonnées à disposition des utilisateurs. Vous créez des packs en fonction de vos groupes d'utilisateurs. Les packs doivent contenir toutes les informations dont un utilisateur ou groupe précis a besoin pour créer des rapports.

Les tâches suivantes sont impliquées lors de la publication de packs :

- Vérifier le modèle et corriger les éventuels problèmes.
- Analyser le modèle à la recherche de problèmes potentiels de modélisation que vous devez traiter avant de publier les métadonnées.
- Définir la sécurité pour restreindre l'accès aux métadonnées et aux données dans les produits IBM Cognos.
- Indiquer les langues publiées avec chaque pack.
- Créer ou modifier un pack.
- Indiquer les options de suppression qui seront disponibles pour les utilisateurs du pack.
- Publier le pack.
- Vérifier le pack et corriger les éventuels problèmes.

Lors de la publication des packs, vous avez la possibilité d'externaliser des sujets de requête et des dimensions pour les convertir à des formats que vous pouvez utiliser dans IBM Cognos Transformer ou dans d'autres applications.

**Remarque :** L'externalisation de requêtes ne sera plus prise en charge dans les versions ultérieures de Transformer.

---

## Vérification des modèles ou des packs

A tout moment du processus de modélisation, vous pouvez vérifier la validité de la totalité du modèle ou des objets sélectionnés dans le modèle, par exemple, un pack. D'abord, vérifiez le modèle et corrigez les éventuels problèmes, puis vérifiez individuellement chaque pack dans le modèle. En vérifiant un pack, vous pouvez trouver et supprimer des objets non valides pouvant entraîner l'échec d'une requête.

Lorsque vous vérifiez un modèle ou un pack, IBM Cognos Framework Manager recherche les messages dans les catégories suivantes :

Catégorie	Description
Incohérences du modèle interne	S'assure que les objets sont définis correctement et que des noms ne sont pas utilisés en double. Toujours exécuter ce groupe de tests.
Références d'objet non valides ou incomplètes	Détecte les références ambiguës et les références à des objets manquants.
Déterminants complètement définis	S'assure que les déterminants sont complètement définis à l'aide de clés et d'attributs et que tous les éléments de requête sont désignés par un déterminant.

Catégorie	Description
Dimensions complètement définies	S'assure que les dimensions sont complètement définies à l'aide de légendes de membre de niveau et de clés métier.
Statut de requête	<p>S'assure que le statut d'évaluation des dimensions et des sujets de requête est valide et qu'il n'est pas défini à "doit être réévalué" ou "non valide".</p> <p>Statut de requête ne vérifie pas les objets de raccourci. Par exemple, vous créez un raccourci vers un objet. L'objet devient non valide pour une raison quelconque. Si vous vérifiez l'objet, un message s'affiche car il n'est pas valide. Lorsque vous vérifiez le raccourci, aucun message n'est affiché.</p>
Compatibilité avec les versions antérieures	Indique la présence dans un modèle de fonctions d'une édition antérieure qui ont été conservées mais qui seront périmées dans une édition future.
Vérification de la cardinalité des relations	Signale l'existence de relations multivoques.

Chaque catégorie peut générer plusieurs messages pour un objet. Pour chaque message, sont fournies la gravité, l'icône d'objet, l'identificateur d'objet, une description du message, des explications sur la façon de corriger le problème et des actions possibles pouvant être effectuées sur l'objet. Vous pourrez peut-être remédier immédiatement à un problème en modifiant l'objet qui a provoqué le problème ou en demandant à Framework Manager de corriger le problème.

## Procédure

1. Effectuez l'une des opérations suivantes :

Objectif	Action
Vérifier un modèle	Dans le menu <b>Projet</b> , cliquez sur <b>Vérifier le modèle</b> .
Vérifier un pack	Dans l' <b>afficheur de projet</b> , cliquez avec le bouton droit de la souris sur un pack et cliquez sur <b>Vérifier les objets sélectionnés</b> .
Vérifier les objets sélectionnés	Dans l' <b>afficheur de projet</b> , sélectionnez un ou plusieurs objets, cliquez avec le bouton droit de la souris, puis cliquez sur <b>Vérifier les objets sélectionnés</b> .

2. Dans l'onglet **Options**, sélectionnez les niveaux de gravité des messages que vous souhaitez afficher dans les résultats.  
Par défaut, toutes les cases de niveau de gravité des messages sont cochées. Désélectionnez les gravités que vous ne souhaitez pas afficher.
3. Sélectionnez les catégories à vérifier.  
Par défaut, toutes les cases de catégorie sont cochées. Désélectionnez les catégories que vous ne souhaitez pas afficher.
4. Cliquez sur **Vérifier le modèle**.  
L'onglet **Vérifier les résultats du modèle** affiche les résultats.

5. Pour trier les messages, cliquez sur **Trier** dans l'en-tête de colonne de gravité, de type d'objet ou de nom.
6. Pour voir l'objet associée au message, sous l'en-tête **Actions**, cliquez sur **Rechercher dans la vue de projet**.
7. Pour réparer des problèmes pour un groupe, sélectionnez un critère de groupement dans la liste.
8. Cochez la case en regard de chaque message correspondant au problème à résoudre.

**Astuces :**

- Pour sélectionner tous les messages, cochez la case située en haut de la colonne de cases à cocher. Pour désélectionner tous les messages sélectionnés, décochez la case en haut de la colonne de cases à cocher.
  - Si vous avez regroupé les messages, cochez la case située en haut de la colonne de cases à cocher du groupe pour sélectionner tous les messages du groupe.
  - Pour masquer les informations détaillées sur les messages, cliquez sur **Réduire** dans l'en-tête du groupe.
9. Cliquez sur **Réparer les objets sélectionnés**.

Framework Manager répare les problèmes dans l'ordre suivant :

- Références non valides
- Objets non valides, relations non valides, règles d'agrégation non valides, paramètres manquants, types d'invite non pris en charge, rôles déduits et réévaluation sans suivre un ordre particulier
- Erreurs de modèle mis à niveau

Vous pouvez également réparer des problèmes en cliquant sur l'icône **Editer la définition** sous l'en-tête **Actions** du message. Dans la boîte de dialogue, modifiez la définition de l'objet si nécessaire.

## Résultats

Lorsque le processus de réparation est terminé, un récapitulatif des résultats de la réparation s'affiche. Le modèle ou le pack est à nouveau vérifié et les résultats s'affichent dans l'onglet **Vérifier les résultats du modèle**.

**Conseil :** Les problèmes qui ne peuvent être résolus lors de la vérification d'un pack peut être réparés à l'aide de la vérification de modèle.

Pour les filtres autonomes, s'il se peut que les objets sous-jacents ne soient pas valides, le message "Réévaluation requise" s'affiche. L'option Réparer ne fonctionne pas pour les filtres autonomes. Pour vous assurer que les filtres autonomes sont valides, ouvrez chaque filtre et enregistrez-le.

---

## Création ou modification de packs

Vous créez un pack pour mettre des métadonnées à disposition des utilisateurs. Un pack est un sous-ensemble d'un projet. Il doit contenir toutes les informations dont un utilisateur ou un groupe spécifique a besoin pour créer des rapports. Vous pouvez également appliquer un système de sécurité au pack.

Par exemple, si votre source de données contient des informations relatives à différents domaines d'une entreprise, vous pouvez décider de créer des packs différents pour les ressources humaines et les finances. Veillez à ce que le pack

réponde à des besoins de génération de rapports exhaustifs mais liés. Chaque rapport peut contenir des informations d'un seul pack.

Lorsque vous créez ou modifiez des packs, tenez compte des points suivants :

- Réutilisation des packs

Vous pouvez réutiliser des packs en créant des packs imbriqués. Lorsque vous créez des packs imbriqués, vous créez un pack principal basé sur d'autres packs existants. L'utilisation de packs imbriqués vous permet de gagner du temps et de faciliter la gestion. L'utilisation de packs imbriqués a pour autre avantage que vous publiez uniquement le pack principal.

Par exemple, vous créez trois packs distincts appelés Canada, Mexique et Etats-Unis. Chaque pack contient les objets du projet et la sécurité appropriés pour ce pack. Vous pouvez créer un pack principal nommé Amérique du Nord et y inclure les packs Canada, Mexique et Etats-Unis.

Vous pouvez également réutiliser des packs pour créer un pack consolidé avec des connexions vers plusieurs sources de données.

- Sélection, masquage ou désélection d'objets

Lorsque vous créez un pack, vous pouvez choisir si les objets d'un projet peuvent être sélectionnés en fonction des besoins des utilisateurs.

Option	Description
Sélectionner	L'objet peut être utilisé dans des rapports et sélectionné par les utilisateurs.
Masquer	<p>Les données contenues dans l'objet ne peuvent pas être utilisées dans des rapports car elles ne peuvent pas être sélectionnées par les utilisateurs.</p> <p>Par exemple, vous incluez un sujet de requête de modèle dans un pack. Comme les sujets de requête de modèle sont dépendants des sujets de requête de source de données («Sujets de requête», à la page 76), vous devez ajouter le sujet de requête de source de données à votre pack. Si vous ne souhaitez pas que les utilisateurs puissent voir le sujet de requête de source de données, masquez-le.</p>
Désélectionner	L'objet n'est pas publié. Il ne peut pas être utilisé dans des rapports et sélectionné par les utilisateurs.

**Remarque :** IBM Cognos Framework Manager prend en charge les fonctionnalités Ctrl+Maj et Alt+Maj. Utilisez ces combinaisons de touches pour sélectionner plusieurs objets à inclure ou masquer. Par exemple, si vous voulez inclure uniquement deux éléments d'une grande branche, sélectionnez-la totalité de celle-ci, puis utilisez la combinaison de touches Ctrl+Maj pour désélectionner les éléments à inclure et masquer les éléments sélectionnés restants.

- Inclusion d'un sujet de requête de modèle dans un pack

Si un sujet de requête de modèle fait référence à d'autres sujets de requête dans une macro ou une invite, veillez à inclure les sujets de requête référencés dans le pack.

Cela peut se produire dans les cas suivants :

- Une macro pour le sujet de requête de modèle fait référence à des éléments de requête d'un autre sujet de requête.
- Un autre sujet de requête est référencé dans les propriétés **Informations d'invite**.

Une fois qu'un pack est publié sur le serveur, il est disponible pour les utilisateurs.

## Création d'un pack

### Procédure

1. Cliquez sur le dossier **Packs** et, depuis le menu **Actions**, cliquez sur **Créer, Pack**.
2. Dans la page **Indiquer un nom**, saisissez le nom du pack et éventuellement une description et une infobulle. Cliquez sur **Suivant**.
3. Indiquez si vous incluez des objets issus de packs existants ou du projet, puis spécifiez les objets que vous souhaitez inclure.  
Si vous avez créé d'autres packs, il est conseillé d'ajouter des références de packs en cliquant sur **Utilisation de packs existants**.
4. Indiquez s'il faut utiliser les droits d'accès par défaut pour le pack :
  - Pour accepter les droits d'accès par défaut, cliquez sur **Terminer**.
  - Pour définir les droits d'accès, cliquez sur **Suivant**.
5. Indiquez qui est autorisé à accéder au pack, puis cliquez sur **Suivant**.  
Vous pouvez ajouter des utilisateurs, des groupes ou des rôles.
6. Déplacez la langue à inclure dans le pack vers la zone **Langues sélectionnées** et cliquez sur **Suivant**.
7. Déplacez les ensembles de fonctions de source de données que vous voulez rendre disponibles dans le pack vers la zone **Ensembles de fonctions sélectionnés**.  
Si l'ensemble de fonctions pour votre fournisseur de source de données n'est pas disponible, assurez-vous qu'il a bien été ajouté au projet. Pour plus d'informations, voir «Sélection d'ensembles de fonctions», à la page 321.
8. Cliquez sur **Terminer** et indiquez si vous souhaitez publier le pack.

## Modification d'un pack

### Procédure

1. Cliquez sur le pack que vous souhaitez modifier.
2. Cliquez sur **Actions, Editer la définition**.
3. Cliquez sur les objets à ajouter ou supprimer dans le pack.  
**Astuce** : Pour basculer entre les options pour un objet, cliquez sur l'icône de l'objet ou sélectionnez une option dans la liste.
4. Cliquez sur **OK**.
5. Si vous souhaitez ajouter ou supprimer des références de pack dans le pack que vous modifiez, cliquez sur **Editer**.

---

## Sécurité

Dans IBM Cognos Framework Manager, les mécanismes de sécurité constituent un moyen de restreindre l'accès aux métadonnées et aux données dans les produits IBM Cognos.

Il existe différents types de sécurité dans Framework Manager :

- Sécurité des données  
Vous pouvez créer un filtre de sécurité et l'appliquer à un sujet de requête spécifique. Le filtre contrôle les données affichées pour les utilisateurs lorsqu'ils configurent leurs rapports.
- Sécurité des objets  
Vous pouvez sécuriser directement un objet en permettant ou en refusant à des utilisateurs d'accéder à celui-ci ou en le laissant masqué pour d'autres utilisateurs.
- Sécurité des packs  
Vous pouvez appliquer un mécanisme de sécurité à un pack et identifier qui a accès à celui-ci.

Chaque type de sécurité utilise des utilisateurs, des groupes et des rôles pour définir l'accès.

Il existe des raisons professionnelles dictant un accès restreint aux données. Par exemple, seuls des utilisateurs spécifiques ont le droit d'accéder aux données confidentielles. Vous pouvez disposer de toutes sortes de données, les utilisateurs n'ayant besoin d'extraire des données qu'à partir de tables ou de colonnes spécifiques. Vous pouvez également disposer d'une table contenant de nombreux enregistrements, les utilisateurs ne devant extraire qu'un sous-ensemble d'enregistrements à partir de cette table.

Si vous utilisez des métadonnées SAP BW, il peut exister une sécurité SAP BW sous-jacente qui affecte l'accès des utilisateurs à des membres du niveau. Vous ne pouvez pas redéfinir la sécurité SAP BW dans Framework Manager. Pour plus d'informations, voir «Importation à partir d'une source de données SAP BW», à la page 197.

Avant d'ajouter un mécanisme de sécurité dans Framework Manager, vérifiez que la sécurité a été correctement configurée dans IBM Cognos BI. Pour plus d'informations, reportez-vous au *Guide d'administration et de sécurité*.

## Utilisateurs, groupes et rôles

Les utilisateurs et les groupes sont créés à des fins d'authentification et d'autorisation. Vous pouvez créer vos propres utilisateurs et groupes dans IBM Cognos BI ou utiliser les utilisateurs et les groupes créés dans d'autres fournisseurs d'authentification.

Pour plus d'informations sur la sécurité, les utilisateurs, les groupes et les rôles, voir le *Guide d'administration et de sécurité*.

### Utilisateurs

Une entrée d'utilisateur est créée et gérée dans un fournisseur d'authentification afin d'identifier de façon unique un compte humain ou informatique. Vous ne pouvez pas créer d'utilisateurs dans IBM Cognos BI.

Les informations sur les utilisateurs, telles que les noms de famille et les prénoms, les mots de passe, les ID, les paramètres régionaux et les adresses électroniques, sont stockées dans les fournisseurs.

Les utilisateurs peuvent devenir membres de groupes définis dans des fournisseurs d'authentification et de groupes définis dans IBM Cognos BI. Un utilisateur peut appartenir à un ou plusieurs groupes. Si des utilisateurs sont membres de plusieurs groupes, leurs droits d'accès sont fusionnés.

## Groupes et rôles

Voici quelques exemples de groupes : Employés, Développeurs ou Personnel de vente. Les membres de groupes peuvent être des utilisateurs et d'autres groupes. L'appartenance à un groupe fait partie de l'identité de base de l'utilisateur. Lorsque les utilisateurs se connectent, ils ne peuvent pas sélectionner un groupe qu'ils souhaitent utiliser pour une session. Ils se connectent toujours avec tous les droits associés aux groupes auxquels ils appartiennent.

Un rôle est un groupe spécial. Il représente une collection d'utilisateurs qui ont des responsabilités et des privilèges semblables dans l'organisation. Les membres de rôles peuvent être des utilisateurs, des groupes et d'autres rôles. L'appartenance à un rôle ne fait pas partie de l'identité de base de l'utilisateur.

Vous pouvez utiliser des groupes créés par votre organisation dans les fournisseurs d'authentification ou créer de nouveaux groupes dans l'espace-noms Cognos.

Créez des groupes IBM Cognos dans les cas suivants :

- Vous ne pouvez pas créer de groupes dans votre fournisseur d'authentification.
- Des groupes dont l'étendue porte sur plusieurs espaces-noms sont requis.
- Des groupes portables pouvant être déployés sont requis.
- Vous souhaitez répondre à des besoins spécifiques d'IBM Cognos Administration.
- Vous voulez éviter d'encombrer les systèmes de sécurité de votre organisation avec des informations utilisées uniquement dans le logiciel IBM Cognos BI.

## Ajout de sécurité des données

Vous pouvez limiter les données représentées par les sujets de requête dans un projet en créant un filtre de sécurité. Le filtre de sécurité contrôle les données affichées pour les utilisateurs lorsqu'ils configurent leurs rapports.

Par exemple, votre équipe commerciale se compose d'un directeur commercial et de quatre responsables des ventes. Vous créez un filtre de sécurité qui inclut un groupe pour le directeur commercial et un groupe pour les responsables des ventes, et vous appliquez le filtre au sujet de requête Salaire. Lorsque le pack est disponible pour les utilisateurs et qu'un rapport est généré pour les responsables des ventes et le directeur commercial, seul le directeur commercial peut afficher les informations salariales des responsables des ventes. Pour plus d'informations sur les groupes, voir «Utilisateurs, groupes et rôles», à la page 260.

Si un utilisateur détient plusieurs rôles, les filtres de sécurité appartenant à ces rôles sont joints avec des clauses OR. Si un rôle est basé sur un autre rôle, les filtres de sécurité sont joints avec des clauses AND.

Vous pouvez baser le filtre de sécurité sur un filtre de sécurité existant. Si vous choisissez cette option, le filtre de sécurité hérite du filtre et de toutes les propriétés du filtre.

Lorsque vous créez un filtre de sécurité, vous pouvez également utiliser des filtres de projet existants ou créer de nouveaux filtres à l'aide de l'éditeur d'expression. Pour plus d'informations, voir «Création de filtres», à la page 155.

## Procédure

1. Cliquez sur le sujet de requête approprié.
2. Cliquez sur **Actions, Spécifier la sécurité des données**.
3. Pour ajouter des utilisateurs, des groupes ou des rôles, procédez comme suit :
  - Cliquez sur **Ajouter des groupes**.
  - Dans la fenêtre **Sélection des utilisateurs et groupes**, ajoutez des utilisateurs, des groupes ou des rôles. Pour plus d'informations à ce sujet, reportez-vous au *Guide d'administration et de sécurité*.
  - Dans la fenêtre **Sélection des utilisateurs et groupes**, cliquez sur **OK**.
4. Si vous souhaitez baser le groupe sur un groupe existant, cliquez sur le groupe dans la colonne **Dépendant de**.

**Astuce** : Si vous ne voyez pas le groupe souhaité dans la liste, vous devez ajouter ce groupe au filtre de sécurité.
5. Si vous souhaitez ajouter un filtre à un groupe, dans la colonne **Filtre**, cliquez sur **Créer/Editer le filtre intégré** ou sur **Insérer à partir du modèle**.

## Fonction macro CSVIdentityName

Si vous souhaitez une sécurité au niveau de la ligne basée sur des valeurs UserClass stockées dans votre source de données, implémentez une table de mappage des paramètres qui mappe les valeurs de la source de données aux rôles et groupes correspondants en fonction de l'ID utilisateur sous lequel vous êtes connecté.

Pour ce faire, utilisez une table de mappage des paramètres comme argument avec la fonction macro CSVIdentityName. Cette fonction macro extrait les informations de compte, de groupe et de rôle pour l'utilisateur en cours. Elle renvoie une chaîne de valeurs séparées par des virgules issues de la table de mappage des paramètres, placées entre apostrophes, par exemple, 'assistants', 'techniciens' ou 'dactylographes'.

La fonction macro CSVIdentityName est utilisée comme clé dans la mappe indiquée. Vous pouvez utiliser la liste renvoyée pour générer des clauses In partielles ou pour filtrer les données en fonction du nom d'identité de l'utilisateur en cours.

Par exemple, vous disposez de classes d'utilisateur dont les noms ne correspondent pas à la table de mappage des paramètres Roles\_Groups :

Clé (rôle ou groupe)	Valeur (classes d'utilisateur)
Tous	Group1
Auteurs	Group2
Administrateurs système	Group3
Utilisateurs de requêtes	Group2
NTLM	Group2



Vous disposez de ce sujet de requête :  
(Security\_column, value 1, value 2, value 3)

Lorsque vous ajoutez un filtre au sujet de requête, le filtre utilise une macro pour rechercher une liste de valeurs, par exemple :

```
Security_column in (#CSVIdentityName(%Roles_Groups)#)
```

Pour les utilisateurs avec le rôle Tous, Auteurs et Administrateurs système, le test se présente comme suit :

```
Security_column in ('Group1','Group2','Group3')
```

### **Fonction macro CSVIdentityNameList**

Si la sécurité des données dans la source de données est identique pour les rôles et les groupes définis dans IBM Cognos BI, vous pouvez utiliser la fonction macro CSVIdentityNameList. La fonction macro accepte éventuellement comme paramètre une liste délimitée par des séparateurs et renvoie alors une liste délimitée par des séparateurs pouvant être utilisée dans un filtre avec l'opérateur In. Vous n'avez pas besoin d'une table de mappage des paramètres.

Voici un exemple :

```
Security_column in (#CSVIdentityNameList()#)
```

Pour les utilisateurs avec le rôle Tous, Auteurs et Administrateurs système, le test se présente comme suit :

```
Security_column in ('Everyone','Authors','System  
Administrators')
```

Tenez compte des points suivants :

- Les utilisateurs peuvent appartenir à plusieurs groupes ou rôles.
- Il n'existe aucun moyen de différencier les groupes et les rôles. Chaque groupe ou rôle doit donc avoir un nom unique.
- Cette fonction ne fonctionne que dans un filtre et renvoie toujours les valeurs 0..n.

## **Sécurité des objets**

La sécurité des métadonnées peut être appliquée directement aux objets d'un projet.

Si vous ne définissez pas une sécurité orientée objets, tous les objets de votre projet sont visibles par quiconque dispose d'un accès au pack. Les utilisateurs, groupes ou rôles auxquels aucun paramètre d'autorisation ou de refus n'est associé pour l'accès à un objet sont considérés comme étant indéfinis. L'objet hérite alors de la sécurité qui a été définie pour son objet parent. Lorsque vous accordez ou refusez explicitement l'accès à un objet, vous remplacez les paramètres hérités. Les objets parent et enfant ont alors des paramètres différents. Lorsque vous appliquez une sécurité à un objet parent, elle est également appliquée à tous les objets enfant de cet objet parent dans le modèle. Une fois que vous avez défini la sécurité pour un objet, vous devez effectuer cette même opération pour tous les objets. Pour cela, vous pouvez paramétrer la sécurité sur l'espace-noms racine.

Vous souhaitez peut-être qu'un objet soit visible uniquement pour certains utilisateurs, groupes ou rôles. Par exemple, dans votre projet, vous pouvez avoir un sujet de requête Salaire. Vous pouvez rendre le sujet de requête Salaire visible pour le groupe Gestionnaires et le cacher pour tous les autres utilisateurs.

Si un utilisateur est membre de plusieurs groupes d'utilisateurs et qu'un objet est visible pour un groupe d'utilisateurs et refusé pour l'autre, l'utilisateur ne peut pas accéder à l'objet. Par exemple, Jane appartient à deux groupes d'utilisateurs : Ventes et Marketing. Le groupe Ventes est autorisé à accéder aux sujets de requête Produits et Ventes, mais pas au sujet de requête Prévisions des ventes. Le groupe Marketing est autorisé à accéder aux sujets de requête Produits, Ventes et Prévisions des ventes. Jane n'est pas autorisée à accéder au sujet de requête Prévisions des ventes.

Lorsque vous sécurisez un objet, un pack est automatiquement créé dans IBM Cognos Framework Manager. Le nom de pack se compose d'un trait de soulignement (\_) et du nom de l'objet sécurisé. Ces packs basés sur des objets sont visibles dans l'**Explorateur**. Vous pouvez utiliser ce pack pour voir quels objets du projet sont inclus, masqués ou exclus d'un groupe d'utilisateurs spécifique.

Chaque fois que vous incluez cet objet dans un pack et que vous le publiez pour vos utilisateurs, les mêmes règles de sécurité sont appliquées à cet objet. Lorsque vous publiez un pack qui contient des objets sécurisés, les objets visibles pour les utilisateurs se trouvent à l'intersection de la définition de pack et des paramètres de sécurité de l'objet. Si la sécurité orientée objets n'est pas utilisée, la sécurité appliquée à un pack reste inchangée.

## **Portée de la sécurité des objets**

La sécurité que vous spécifiez pour un objet est transmise aux raccourcis qui font référence à l'objet sécurisé. Si vous disposez d'un raccourci vers un objet sécurisé, le raccourci n'est visible dans le pack publié que pour les utilisateurs autorisés à voir cet objet sécurisé.

Si un sujet de requête de modèle, un calcul ou un filtre fait référence à un objet sécurisé, la sécurité de l'objet n'est pas transmise au sujet de requête de modèle, au calcul, ni au filtre.

Lorsque vous créez un pack contenant le raccourci, l'objet sécurisé n'a pas besoin d'être inclus dans le pack.

Par exemple, seuls les directeurs commerciaux sont autorisés à voir le sujet de requête Objectifs de ventes. Vous créez un raccourci vers le sujet de requête Objectifs de ventes. Lorsque vous conditionnez le modèle, vous incluez le raccourci mais pas le sujet de requête Objectifs de ventes. Seuls les directeurs commerciaux sont habilités à voir le raccourci dans le pack publié.

Si votre modèle est segmenté, la sécurité des objets n'est pas héritée du modèle principal. Vous devez définir la sécurité des objets sur tous les segments de modèle.

## **Ajout de la sécurité orientée objets**

Pour implémenter la sécurité orientée objets, vous devez ajouter un utilisateur, un groupe ou un rôle spécifique à l'objet. Seuls ces utilisateurs, groupes ou rôles peuvent accéder à l'objet.

Les conseils suivants peuvent s'avérer utiles lors de l'ajout d'une sécurité orientée objets :

- Pour afficher la liste des packs basés sur des objets, cliquez deux fois sur le dossier **Packs**. La liste s'affiche dans l'onglet **Explorateur**. Pour voir quels objets

sont sécurisés pour un pack basé sur des objets spécifique, cliquez sur le dossier **Packs** puis, dans le menu **Actions**, cliquez sur **Packs, Explorer les packs** («Exploration de packs», à la page 275).

- Pour déterminer si la sécurité orientée objets est définie dans le modèle, cliquez sur le dossier **Packs** puis, dans le menu **Actions**, cliquez sur **Packs, Explorer les packs**. Cliquez sur l'onglet de l'**explorateur de rôles**. Si la sécurité orientée objets a été définie, un pack s'affiche pour le rôle Tous.
- Pour identifier les objets qui sont explicitement sécurisés dans le modèle, examinez l'icône de l'objet dans l'**afficheur de projet**. L'angle supérieur gauche de l'icône comporte une superposition.
- Pour rechercher tous les objets qui ont été explicitement sécurisés sous un objet donné, sélectionnez l'objet et cliquez sur **Outils, Rechercher tous les objets sécurisés**.

### Procédure

1. Cliquez sur l'objet à sécuriser puis, dans le menu **Actions**, cliquez sur **Spécifier la sécurité de l'objet**.

**Astuce :** Vous pouvez sélectionner plusieurs objets à la fois.

2. Sélectionnez les utilisateurs, les groupes ou les rôles à modifier. Ou cliquez sur **ajouter** pour ajouter des utilisateurs, des groupes ou des rôles.

Pour plus d'informations, reportez-vous au *Guide d'administration et de sécurité*.

3. Indiquez des droits de sécurité pour chaque utilisateur, groupe ou rôle en effectuant l'une des opérations suivantes :

- Pour refuser l'accès à un utilisateur, un groupe ou un rôle, cochez la case **Refuser l'accès** pour celui-ci. Refuser l'accès a la priorité sur Autoriser.
- Pour accorder l'accès à un utilisateur, un groupe ou un rôle, cochez la case **Autoriser** pour celui-ci.

**Astuce :** Pour autoriser tout le monde à afficher tous les objets, sauf si l'accès spécifiquement est refusé, cochez la case **Autoriser** pour le rôle Tous.

4. Cliquez sur **OK**.

La liste des packs basés sur des objets nouveaux et mis à jour s'affiche.

### Suppression de la sécurité orientée objets pour un objet

Pour supprimer la sécurité orientée objets pour un utilisateur, un groupe ou un rôle particulier, supprimez le pack pour celui-ci dans l'**afficheur de projet**.

Pour supprimer complètement la sécurité orientée objets du modèle, supprimez le pack pour le rôle Tous dans l'**afficheur de projet**. Vous ne pouvez pas refuser au groupe **Tous** l'accès à tous les objets.

### Procédure

1. Cliquez sur l'objet requis.
2. Cliquez sur **Actions, Spécifier la sécurité de l'objet**.
3. Supprimez des droits de sécurité en désélectionnant les cases **Autoriser** et **Refuser** pour tous les utilisateurs, groupes ou rôles.
4. Cliquez sur **OK**.

La liste des packs affectés par ces modifications s'affiche.

## Modification de la sécurité d'un pack

Vous pouvez utiliser IBM Cognos Connection pour définir ou modifier la sécurité des métadonnées après qu'un pack a été publié.

Les paramètres de sécurité modifiés par le biais d'IBM Cognos Connection sont ajoutés à la définition de sécurité du portail du pack. Les paramètres affectent l'accès, mais ne modifient pas la définition de pack dans IBM Cognos Framework Manager.

Pour définir la sécurité des métadonnées la première fois que vous publiez un pack, voir «Publication de packs», à la page 269.

Vous pouvez organiser la sécurité en spécifiant les utilisateurs, groupes et rôles qui ont accès à certaines parties du modèle publié.

Pour ajouter un mécanisme de sécurité aux métadonnées, procédez comme suit :

- Décidez si les objets peuvent être sélectionnés, désélectionnés ou masqués dans le pack.
- Sélectionnez les utilisateurs qui disposeront d'un accès administrateur à un pack.
- Ajoutez des utilisateurs, des groupes et des rôles au pack.

Lorsque vous appliquez un accès administrateur à un pack, vous accordez l'accès à ou aux utilisateurs chargés de

- republier un pack de Framework Manager sur le serveur IBM Cognos,
- s'assurer qu'aucun rapport n'est affecté lorsqu'un pack Framework Manager est republié sur le serveur.

### Procédure

1. Cliquez sur le pack à modifier, puis, dans le menu **Actions**, cliquez sur **Pack, Éditer les paramètres du pack**.

Cette procédure appelle IBM Cognos Connection.

2. Dans IBM Cognos Connection, cliquez sur l'onglet **Droits**.
3. Créez, ajoutez ou supprimez des groupes ou des rôles selon les besoins. Pour plus d'informations à ce sujet, reportez-vous au *Guide d'utilisation* d'IBM Cognos Connection.
4. Une fois que vous avez fini de modifier la définition de sécurité pour le pack, cliquez sur **OK** pour revenir dans Framework Manager.

---

## Spécification de langues

Vous pouvez indiquer les langues publiées avec chaque pack. Vous pouvez créer plusieurs packs basés sur le même modèle, chacun utilisant une langue différente.

Par exemple, le pack pour le bureau de vente mexicain inclut l'espagnol et l'anglais. Le pack pour le bureau de vente canadien comprend le français et l'anglais.

Vous pouvez également spécifier les langues pour tous les packs à la fois.

Vous devez ajouter des langues au projet («Métadonnées multilingues», à la page 125) pour pouvoir spécifier les langues requises par les utilisateurs dans les packs.

## Spécification de langues pour un pack

### Procédure

1. Dans l'**afficheur de projet**, cliquez sur le pack à modifier.

2. Dans l'onglet **Propriétés**, recherchez la propriété **Langue**, puis cliquez sur **Cliquez pour éditer**.
3. Cliquez sur une langue (ou appuyez sur Ctrl et cliquez sur plusieurs langues) dans la zone **Langues de projet disponibles**, puis utilisez la flèche pour la déplacer vers la zone **Langues sélectionnées**.

## Spécification de langues pour tous les packs

### Procédure

1. Dans l'afficheur de **projet**, cliquez sur le dossier **Packs**.
2. Dans le menu **Actions**, cliquez sur **Packs, Spécifier les langues de pack**.
3. Cochez la case correspondant à la langue souhaitée pour chaque pack.

---

## Définition d'options de suppression

Vous pouvez définir des propriétés de suppression pour le pack publié avec IBM Cognos Framework Manager. Ces propriétés déterminent si les utilisateurs des studios IBM Cognos peuvent sélectionner la suppression sur plusieurs arêtes ou sur une seule arête. Elles déterminent aussi les types de valeurs pouvant être supprimées. Les types de valeurs que les utilisateurs peuvent choisir de supprimer dépendent du studio.

Lorsqu'un pack est créé, les propriétés de suppression sont automatiquement définies sur **true**. Si vous souhaitez modifier les valeurs sur un pack publié, vous devez republier le pack.

### Procédure

1. Dans l'afficheur de **projet**, cliquez sur le pack à modifier.
2. Dans l'onglet **Propriétés**, recherchez la propriété de suppression, puis sélectionnez l'une des valeurs suivantes.
  - **Permettre la suppression des valeurs nulles** - Avec **true**, cette propriété rend la suppression disponible pour les utilisateurs des studios IBM Cognos. Avec **false**, la suppression n'est pas disponible dans le pack publié.
  - **Permettre la suppression sur plusieurs arêtes** - Avec **true**, les utilisateurs peuvent sélectionner des options de suppression sur plusieurs arêtes ou sur une seule arête. Avec **false**, les utilisateurs ont uniquement accès à la suppression sur une seule arête. La propriété **Permettre la suppression des valeurs nulles** doit également avoir la valeur **true**.
  - **Permettre l'accès aux options de suppression** - Avec la valeur **true**, les utilisateurs peuvent sélectionner les types de valeurs qui seront supprimés, comme zéro ou les valeurs manquantes. Par défaut, tous les types de valeurs sont supprimés. La propriété **Permettre la suppression des valeurs nulles** doit également avoir la valeur **true**.

---

## Dimensions et sujets de requête externalisés

Lorsque vous publiez un pack, vous avez la possibilité d'externaliser les sujets de requête et les dimensions dans des formats utilisables dans d'autres applications IBM Cognos. Des considérations particulières s'appliquent lors de l'externalisation de modèles basés sur des métadonnées SAP BW.

Pour plus d'informations, voir «Utilisation de données SAP BW à l'aide de fichiers CSV externalisés dans Framework Manager», à la page 661.

Vous devez d'abord définir la manière dont chaque objet va être externalisé en spécifiant la méthode à utiliser. Lorsque vous publiez le pack, vous indiquez que les sujets de requête et les dimensions doivent être externalisés.

Si vous avez indiqué un nombre maximal de lignes à extraire dans la boîte de dialogue **Restrictions**, ce paramètre est ignoré.

Il existe plusieurs méthodes d'externalisation.

### **Méthode par défaut**

Utilisez la méthode par défaut pour spécifier les objets d'un pack qui ne doivent pas être externalisés. Pour améliorer les performances, vous pouvez ne pas externaliser tous les objets d'un pack.

### **Méthode TAB**

Utilisez la méthode TAB pour générer un fichier délimité par des tabulations contenant les résultats du sujet de requête ou de la dimension. Le fichier généré peut être utilisé directement comme une source de données. Il contient des données basées sur le codage Unicode utilisant UTF-16 LE (Little Endian) avec BOM (marque d'ordre d'octet). Un fichier est généré pour chaque sujet de requête ou dimension qui est configuré pour être externalisé.

Cette méthode ne fonctionne pas avec Transformer car ce programme ne prend pas en charge Unicode. Utilisez la méthode CSV pour créer des fichiers pour Transformer.

### **Propriété Externaliser le récapitulatif automatique**

Vous pouvez spécifier que la sortie doit être agrégée et/ou regroupée. Par défaut, IBM Cognos Framework Manager renvoie des lignes au niveau de détail sans appliquer aucune agrégation ni aucun regroupement. Utilisez cette propriété si vous souhaitez que des données relationnelles soient agrégées lors de leur externalisation. Indiquez des déterminants pour le sujet de requête avant de l'externaliser.

Utilisez la propriété **Externaliser le récapitulatif automatique** pour appliquer la valeur de la propriété **Agrégat ordinaire** aux éléments de requête dont la propriété **Utilisation** a pour valeur **Fait**.

Si vous souhaitez que les éléments figurent dans un ordre spécifique dans la clause Group By, spécifiez d'abord les déterminants, puis définissez la propriété **Externaliser le récapitulatif automatique**.

Vous pouvez utiliser la propriété **Externaliser le récapitulatif automatique** avec toutes les méthodes d'externalisation.

### **Types de données pris en charge**

Framework Manager prend en charge des chaînes, des entiers et des dates. Il ne prend pas en charge les dimensions de temps. Utilisez une clé de date sur le sujet de requête de fait dans Framework Manager, et laissez Transformer générer la dimension de temps.

## Raccourcis

Si un raccourci est inclus dans un pack et qu'il pointe vers un sujet de requête qui a été externalisé, ce raccourci est également externalisé. Le nom du fichier de données correspond au nom du sujet de requête vers lequel pointe le raccourci. Si plus d'un raccourci pointe vers le même sujet de requête, ce dernier est externalisé chaque fois que le raccourci est détecté.

## Traitement de requête

Certaines requêtes provoquent l'émission de plusieurs requêtes et/ou nécessitent un traitement local pour extraire des données. Pour empêcher cela, prenez soin d'attribuer la valeur **Base de données seulement** à la propriété **Traitement de requête** pour tous les objets de source de données dans le modèle. Pour plus d'informations sur le traitement de requête, voir «Optimisation des performances en configurant le type de traitement de requête», à la page 317.

## Processus permettant d'externaliser des dimensions

Pour externaliser les dimensions, exécutez les tâches suivantes :

- Créez un sujet de requête de modèle ou un sujet de requête de source de données contenant les dimensions à externaliser. Pour plus d'informations, voir «Sujets de requête», à la page 76.
- Ajoutez les filtres dont vous avez besoin.  
Pour en savoir davantage sur les filtres des métadonnées relationnelles, voir «Création de filtres», à la page 155. Pour en savoir davantage sur les filtres des métadonnées SAP BW, voir «Création d'un filtre (SAP BW)», à la page 242.
- Dans la sous-fenêtre **Propriétés**, définissez la propriété **Méthode d'externalisation** avec la méthode de votre choix.
- Publiez le pack pour externaliser les dimensions que vous avez sélectionnées.

---

## Publication de packs

Dans IBM Cognos Framework Manager, vous pouvez publier un pack dans un dossier de Content Manager pour que les utilisateurs puissent y accéder.

Vous pouvez également publier un pack dans un emplacement réseau. Un pack dans un emplacement réseau ne peut pas être utilisé par les utilisateurs. La publication dans un emplacement réseau s'avère utile pour la sauvegarde d'un pack. Lorsque vous publiez dans un emplacement du réseau local, veillez à ne pas écraser des fichiers existants, notamment des modèles Framework Manager et le modèle actuellement ouvert.

Les objets qui sont exclus ou masqués, mais qui sont nécessaires dans un chemin de requête, sont inclus mais marqués comme étant masqués dans un pack publié.

Les paramètres de restriction ont la priorité sur les paramètres appliqués au modèle actuellement ouvert (qu'il s'agisse d'un modèle parent ou enfant).

Pour éviter les problèmes, cochez la case **Vérifier le pack avant publication** dans l'assistant de publication pour vous assurer que le pack est complet et ne contient pas d'erreurs.

Lorsque vous publiez un pack, vous pouvez effectuer les opérations suivantes :

- Définir le nombre de versions de modèle à conserver sur le serveur. Pour plus d'informations sur les versions de modèle, voir «Mise à jour de rapports pour utiliser la version la plus récente de packs», à la page 274.  
**Astuce :** Pour afficher le nombre de versions de modèle défini pour un pack, sélectionnez ce dernier et recherchez la propriété **Nombre maximal de versions** dans la sous-fenêtre **Propriété**.
- Externaliser des sujets de requête et des dimensions pour pouvoir les utiliser avec Transformer («Dimensions et sujets de requête externalisés», à la page 267).
- Indiquer si un pack utilise le mode de requête dynamique.

Pour en savoir davantage sur le mode de requête dynamique, reportez-vous au *Guide sur les requêtes dynamiques*.

**Remarque :** Vous pouvez créer des packs directement dans IBM Cognos Connection pour les PowerCubes IBM Cognos, et les cubes et requêtes SAP BW. Pour plus d'informations, voir la section sur les packs du *guide d'administration et de sécurité*.

## Procédure

1. Sélectionnez le pack à publier.
2. Dans le menu **Actions**, cliquez sur **Pack, Publier les packs**.
3. Choisissez un emplacement pour la publication du pack :
  - Pour publier le pack sur le serveur de rapports, cliquez sur **IBM Cognos Content Store, Ouvrir**, puis sélectionnez un dossier existant ou créez un nouveau dossier dans le magasin de contenu.
  - Pour publier le pack dans un emplacement réseau, cliquez sur **Emplacement sur le réseau**. Veillez à sélectionner un emplacement autre que le répertoire dans lequel les fichiers de projet sont stockés. En règle générale, évitez d'enregistrer dans le même emplacement qu'un modèle, car le modèle risque d'être écrasé.
4. Pour activer la gestion des versions de modèle lors de la publication dans IBM Cognos Content Store, cochez la case **Activer la gestion des versions de modèle** et entrez le numéro des versions de modèle du pack à conserver.  
**Astuce :** Pour tout supprimer sauf la dernière version publiée sur le serveur, cochez la case **Supprimer toutes les versions de modèle précédentes**.
5. Cliquez sur **Suivant**.
6. Dans la fenêtre **Ajouter la sécurité**, définissez la sécurité pour le pack (facultatif) :

Objectif	Actions
Créer, ajouter ou supprimer un utilisateur, un groupe ou un rôle.	<p>Dans l'onglet <b>Accès utilisateur</b>, cliquez sur <b>Ajouter</b>.</p> <p>Dans la fenêtre <b>Sélection des utilisateurs et groupes</b>, définissez la sécurité des utilisateurs. Pour plus d'informations sur l'utilisation de la fenêtre <b>Sélection des utilisateurs et groupes</b>, reportez-vous au <i>guide d'administration et de sécurité</i>. Les utilisateurs, les groupes ou les rôles définis dans l'onglet <b>Accès utilisateur</b> disposent de droits de lecture, d'écriture, d'exécution et de passage.</p>



Objectif	Actions
Accorder un accès administrateur à un utilisateur, un groupe ou un rôle.	<p>Dans l'onglet <b>Accès administrateur</b>, cliquez sur <b>Ajouter</b>.</p> <p>Dans la fenêtre <b>Sélection des utilisateurs et groupes</b>, définissez la sécurité des administrateurs. Pour plus d'informations sur l'utilisation de la fenêtre <b>Sélection des utilisateurs et groupes</b>, reportez-vous au <i>guide d'administration et de sécurité</i>. Les utilisateurs, les groupes ou les rôles définis dans l'onglet Accès administrateur disposent de droits de lecture, d'écriture, de définition des règles et de passage.</p>

**Remarque :** La fenêtre **Ajouter la sécurité** de l'assistant de publication est disponible uniquement la première fois que vous publiez un pack. Si vous republiez le pack dans le même emplacement, vous ne pouvez pas remplacer la sécurité existante. Pour modifier la sécurité pour le pack publié, reportez-vous à la section «Modification de la sécurité d'un pack», à la page 265.

7. Cliquez sur **Suivant**.
8. Si vous souhaitez externaliser des sujets de requête, cochez la case **Générer les fichiers pour les sujets de requête externalisés**.
9. Par défaut, le pack est vérifié avant d'être publié. Si vous ne souhaitez pas vérifier votre modèle avant de le publier, désélectionnez la case **Vérifier le pack avant publication**.
10. Si le pack contient des sources de données prises en charge par le mode de requête dynamique, vous avez la possibilité d'activer le mode de requête dynamique pour le pack. Cochez la case **Utiliser le mode de requête dynamique**. Vous êtes invité à confirmer cette sélection lorsque vous cliquez sur **Publier**. Cette étape est applicable lorsque la propriété **Mode requête** est définie sur Compatible.

**Remarque :** Si un pack contenant des sources de données prises en charge et non prises en charge est publié avec le mode de requête dynamique, les utilisateurs recevront une erreur lors de l'ouverture du pack dans les studios. Pour plus d'informations sur les sources de données prises en charge par le mode de requête dynamique, voir le *guide d'administration et de sécurité* d'IBM Cognos.

11. Cliquez sur **Publier**.  
Si vous avez choisi d'externaliser des sujets de requête, Framework Manager répertorie les fichiers qui ont été créés.
12. Cliquez sur **Terminer**.

## Publication de packs basés sur des sources de données OLAP

Vous pouvez utiliser IBM Cognos Framework Manager pour vous connecter à une source de données OLAP et créer un pack basé sur un cube. Vous pouvez ensuite publier le pack directement dans IBM Cognos Connection pour le rendre disponible pour être utilisé avec les studios IBM Cognos.

Par défaut, chaque pack contient une connexion à un seul cube. Si vous souhaitez créer un pack contenant plusieurs cubes, exécutez l'assistant de métadonnées et créez un pack pour chaque cube. Créez ensuite un pack qui inclut des packs individuels selon vos besoins.

Avant de créer un pack contenant plusieurs cubes, prenez en compte l'impact potentiel sur les performances. Dans IBM Cognos Connection, chaque fois qu'un pack est utilisé, une connexion est établie pour chacune des sources de données définies dans le pack. La création de packs de grande taille avec plusieurs cubes peut avoir un impact négatif sur les performances. Pour atténuer la dégradation potentielle des performances dues à la création d'un pack volumineux contenant de nombreux cubes, créez un pack par cube, puis créez des combinaisons de packs plus petites selon les besoins.

Par défaut, les packs basés sur une source de données OLAP ne contiennent pas de listes de fonction de fournisseur. Si vous souhaitez inclure la liste de fonctions du fournisseur, indiquez l'ensemble de fonctions à inclure, puis republiez le pack.

**Remarque :** Pour publier un pack contenant un seul cube, utilisez IBM Cognos Connection. Pour en savoir davantage sur la publication d'un pack à partir d'IBM Cognos Connection, reportez-vous au *Guide d'administration et de sécurité*.

## Procédure de création et de publication d'un pack

### Procédure

1. Dans la page **Bienvenue**, cliquez sur **Créer un nouveau projet**.  
**Astuce :** Si vous êtes déjà dans Framework Manager, cliquez sur **Fichier, Nouveau projet**.
2. Dans la page **Nouveau projet**, spécifiez un nom et un emplacement pour le projet et cliquez sur **OK**.  
Vous pouvez être invité à entrer des informations d'authentification.
3. Dans la page **Sélectionner la langue**, cliquez sur la langue de conception du projet.  
La langue que vous sélectionnez ne peut pas être modifiée une fois que vous avez cliqué sur **OK**, mais vous pouvez ajouter d'autres langues. Pour plus d'informations, voir «Ajout d'une langue à un projet», à la page 129.
4. Dans la boîte de dialogue **Assistant de métadonnées**, cliquez sur **Source de données**, puis sur **Suivant**.
5. Sélectionnez votre source de données dans la liste des connexions de source de données disponibles, puis cliquez sur **Suivant**.  
Si la connexion de source de données n'est pas disponible dans la liste, vous pouvez cliquer sur **Nouveau** pour la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.
6. Indiquez un nom pour le pack et cliquez sur **Suivant**.  
(Facultatif) Vous pouvez indiquer une description et une infobulle pour le pack.
7. Spécifiez qui a accès à ce pack.  
Vous pouvez ajouter des utilisateurs, des groupes ou des rôles. Pour plus d'informations, voir «Utilisateurs, groupes et rôles», à la page 260.
8. Cliquez sur **Terminer** pour importer les métadonnées et créer le pack.
9. Lorsque vous y êtes invité, cliquez sur **Oui** pour publier le pack, ou sur **Non** pour revenir dans l'**afficheur de projet**.

## Résultats

L'espace-noms s'affiche dans l'**afficheur de projet**. Vous ne pouvez pas voir les objets du modèle de métadonnées natif depuis Framework Manager. Les objets de métadonnées natifs sont visibles dans les studios IBM Cognos lorsque le pack de métadonnées natif est utilisé.

## Procédure d'ajout d'un autre pack

### Procédure

1. Dans l'**afficheur de projet**, cliquez avec le bouton droit de la souris sur le modèle à utiliser, puis cliquez sur **Exécuter l'assistant de métadonnées**.
2. Dans la boîte de dialogue **Assistant de métadonnées**, cliquez sur **Sources de données**, puis sur **Suivant**.
3. Sélectionnez votre source de données dans la liste des connexions de source de données disponibles, puis cliquez sur **Suivant**.  
Si la connexion de source de données n'est pas disponible dans la liste, vous pouvez cliquer sur **Nouveau** pour la créer. Pour plus d'informations, voir «Création de connexions de sources de données», à la page 43.
4. Indiquez un nom pour le pack et cliquez sur **Suivant**.  
(Facultatif) Vous pouvez indiquer une description et une infobulle pour le pack.
5. Définissez les droits d'accès pour le pack.  
Vous pouvez ajouter des utilisateurs, des groupes ou des rôles au pack. Pour plus d'informations, voir «Utilisateurs, groupes et rôles», à la page 260.
6. Cliquez sur **Terminer** pour importer les métadonnées et créer le pack.
7. Lorsque vous y êtes invité, cliquez sur **Oui** pour publier le pack, ou sur **Non** pour revenir dans l'**afficheur de projet**.

### Résultats

L'espace-noms s'affiche dans l'**afficheur de projet**. Vous ne pouvez pas voir les objets du modèle de métadonnées natif depuis Framework Manager. Les objets de métadonnées natifs sont visibles dans les studios IBM Cognos lorsque le pack de métadonnées natif est utilisé.

## Publication de packs en exécutant des scripts

Les rapports IBM Cognos sont dépendants des objets du pack sur lequel le rapport est basé. Si votre rapport utilise des objets qui n'existent plus dans le pack, il ne sera pas exécuté. Vous pouvez être confronté à ce problème si vous apportez des modifications à la source de données physique, puis que vous utilisez un script pour générer le modèle et republier le pack. Une fois que le script est exécuté et que le pack est publié, l'analyse de l'impact de la publication du pack ne permet pas d'identifier les rapports altérés.

Pour éviter ce problème lors de l'utilisation de scripts pour publier des packs, effectuez les tâches de publication dans l'ordre indiqué.

### Procédure

1. Exécutez le script en excluant les étapes de publication du pack.
2. Vérifiez le modèle ou analysez l'impact de la publication du pack.
3. Publiez le pack.

## Mise à jour de rapports pour utiliser la version la plus récente de packs

Lorsque vous publiez un pack pour la première fois, vous créez un pack correspondant sur le serveur IBM Cognos. Le pack contient un modèle mais pas de rapports.

Lorsque vous publiez un pack, vous pouvez sélectionner le nombre de versions du modèle à conserver sur le serveur. La fois suivante que vous publiez le pack, la version du modèle est mise à jour dans le pack existant sur le serveur.

Les rapports nouveaux ou modifiés utilisent la dernière version du modèle dans le pack. Lorsqu'un rapport est enregistré, la version du modèle utilisée est enregistrée dans la spécification de rapport. En cas de nouvelle publication du pack, l'auteur du rapport est informé du fait que le rapport utilisera sa dernière version. L'auteur du rapport doit enregistrer le rapport pour achever la mise à jour. Si vous ouvrez un rapport enregistré après la nouvelle publication du pack avec lequel il a été créé, il se produit l'un des deux événements suivants :

- Si la version d'origine du pack existe encore, le rapport est exécuté sur la base de celle-ci. Si vous souhaitez que le rapport soit exécuté avec la dernière version du pack, vous devez mettre à jour le rapport pour qu'il utilise la version la plus récente du pack. Reportez-vous aux étapes ci-dessous.
- Si la version d'origine du pack n'existe plus, le rapport est mis à jour pour s'exécuter sur la base de la version la plus récente.

### Procédure

Effectuez l'une des opérations suivantes :

- Faites passer un rapport vers la dernière version du modèle en éditant, puis en enregistrant le rapport.
- Avant de republier le modèle, faites passer tous les rapports vers la dernière version du modèle en cochant la case **Supprimer toutes les versions de modèle précédentes** dans l'assistant **Publier**.
- Avant de republier le modèle, désactivez la gestion des versions de modèle en définissant la limite de version de modèle sur 1.

---

## Chapitre 8. Gestion de projet

Au cours de la durée de vie d'un projet, les données peuvent être modifiées et de nouvelles exigences peuvent apparaître et vous obliger à mettre à jour les modèles et les sources de données.

Après avoir publié votre projet, vous pouvez procéder comme suit pour gérer le contenu :

- Comprendre les métadonnées contenues dans votre modèle.
- Implémenter une modélisation multiutilisateur.
- Administrer les métadonnées.
- Synchroniser des projets.
- Contrôler et optimiser le comportement des requêtes.

---

### Métadonnées de votre modèle

Avant d'apporter des modifications à un modèle publié, vous pouvez mieux comprendre les métadonnées en procédant comme suit:

- En explorant vos projets pour voir les packs et les rôles d'un projet.
- En affichant les différents objets de vos packs pour voir où ils figurent.
- En créant une documentation relative à votre modèle pour répondre aux besoins spécifiques de votre entreprise.

### Exploration de packs

Si vous disposez d'un grand nombre de projets et de systèmes de sécurité orientés objets dans un projet, il peut être difficile de garder tout bien organisé. Vous pouvez explorer des packs pour afficher les packs et les rôles d'un projet.

Dans l'onglet **Contenu du pack**, vous voyez une liste de tous les packs (normaux et basés sur des objets) d'un projet, ainsi que les objets qui ont été sélectionnés, désélectionnés ou masqués pour chaque pack.

L'onglet **Sécurité d'objet** affiche une liste de tous les utilisateurs, groupes et rôles d'un projet, et indique dans quel pack la sécurité orientée objets est appliquée. Vous pouvez également voir si les objets du projet sont masqués ou visibles pour un utilisateur, un groupe ou un rôle spécifique.

### Procédure

1. Sélectionnez le dossier **Packs**.
2. Dans le menu **Actions**, cliquez sur **Pack, Explorer les packs**.
3. Sélectionnez l'action à exécuter.

Objectif	Action
Afficher le contenu d'un pack	Cliquez sur l'onglet <b>Contenu du pack</b> .
Editer le pack	Cliquez sur l'onglet <b>Contenu du pack</b> , sélectionnez le pack et cliquez sur <b>Editer</b> .  Pour plus d'informations, voir «Création ou modification de packs», à la page 257.

Objectif	Action
Afficher la sécurité de chaque pack	Cliquez sur l'onglet <b>Sécurité d'objet</b> et sélectionnez un pack.

4. Cliquez sur **Fermer**.

## Affichage de la distribution d'objets dans des packs

Lorsque vous affichez l'inclusion d'un objet dans les packs, vous voyez, par pack, où l'objet existe et s'il est sélectionné, désélectionné ou masqué dans ce pack.

Si l'objet est sécurisé, vous verrez également le pack basé sur des objets dans lequel figure l'objet.

### Procédure

1. Cliquez sur l'objet que vous souhaitez visualiser, et cliquez sur **Actions, Editer l'inclusion de pack**.
2. Pour éditer le pack, cliquez sur **Editer le pack**.  
Pour plus d'informations, voir «Création ou modification de packs», à la page 257.
3. Cliquez sur **OK**.

## Création de la documentation du modèle

Une fois que vous avez modélisé les métadonnées, vous pouvez créer une représentation HTML ou XML du modèle qui peut être personnalisée et imprimée. Cela s'avère utile pour le débogage de votre modèle ou si votre société a besoin de ce type de documentation pour répondre à des exigences de traitement.

Lorsque vous créez la documentation du modèle, vous pouvez documenter la totalité du modèle ou vous pouvez sélectionner un sous-ensemble du modèle. Pour documenter le modèle entier, vous devez cliquer sur l'espace-noms de niveau supérieur. La documentation du modèle affiche l'objet sélectionné, ainsi que toutes les propriétés et les enfants de cet objet.

Vous pouvez afficher, enregistrer ou imprimer le rapport au format XML ou HTML. Par défaut, le format XML dans IBM Cognos Framework Manager est toujours le format XML brut. Pour personnaliser la sortie de rapport XML dans votre navigateur, vous pouvez fournir votre propre transformation XSLT.

Pour personnaliser votre sortie HTML, vous pouvez utiliser votre propre transformation XSLT en spécifiant le chemin d'accès et le fichier XSL dans la section ModelDocXSL du fichier fm.ini situé dans le répertoire *emplacement\_installation/configuration*. Framework Manager utilisera le fichier XSL spécifié lors de l'affichage de la version HTML du modèle de rapport.

### Procédure

1. Cliquez sur l'objet à documenter.

**Conseil :** Cliquez sur l'espace-noms de niveau supérieur pour documenter la totalité du modèle.

2. Cliquez sur **Outils, Rapport de modèle**.

Le rapport de modèle s'affiche.

Vous pouvez enregistrer ou imprimer le rapport, ou modifier son format.

---

## Modélisation multiutilisateur

Vous pouvez implémenter la modélisation multiutilisateur dans IBM Cognos Framework Manager par des fonctions de branchement ou de fusion ou de segmentation et de liaison.

- Branchement et fusion

Si vous utilisez les fonctions de branchement et de fusion pour gérer un projet multiutilisateur, chaque utilisateur peut modifier les mêmes objets d'un projet en même temps. Chaque utilisateur dispose d'une copie du projet entier et peut travailler sur n'importe quelle partie de celui-ci. Lorsque les branches sont de nouveau fusionnées dans le projet racine, tous les conflits entre celui-ci et les branches sont résolus. Pour plus d'informations, voir «Branchement et fusion de projets».

- Segmentation et liaison

Si vous utilisez les fonctions de segmentation et de liaison pour gérer un projet multiutilisateur, chaque utilisateur peut visualiser les mêmes parties d'un projet en même temps. Toutefois, vous devez vous assurer que chaque utilisateur modifie des parties individuelles du projet. Utilisez des liens pour permettre aux différents utilisateurs de faire référence à un projet pendant qu'un autre utilisateur travaille dessus. Pour plus d'informations, voir «Projets segmentés et liés», à la page 283.

### Branchement et fusion de projets

Le branchement et la fusion permettent à plusieurs utilisateurs de travailler simultanément sur le même modèle. Pour ce faire, le propriétaire du projet crée une branche du projet racine constituant une copie du projet. Un membre de l'équipe peut modifier la branche si nécessaire, indépendamment du projet racine. Des branches peuvent être à nouveau fusionnées dans le projet racine si besoin est. Les conflits entre le projet racine et une branche sont résolus pendant le processus de fusion.

Le nombre de branches pouvant être créées n'est pas limité. Vous pouvez créer une branche à partir d'une branche.

### Branchement et fusion

Tenez compte des instructions suivantes quand vous effectuez un branchement sur un projet :

- Décidez du mode de partage des métadonnées dans votre organisation. Pour plus d'informations, voir «Méthodologies de branchement», à la page 278.
- Scindez le projet en éléments et branches logiques et effectuez un branchement sur le projet pour créer les éléments logiques. Pour plus d'informations, voir «Méthodes de branchement d'un projet», à la page 280.
- La communication entre les membres d'équipe travaillant sur les différentes branches est essentielle. Avant d'apporter une modification importante à votre branche, parlez-en avec les autres membres de l'équipe afin de voir l'impact que la modification aura sur leurs branches. Par exemple, l'ajout d'objets à une branche n'aura probablement pas d'incidence pour les autres membres de l'équipe. Par contre, la suppression d'un objet que d'autres membres de l'équipe utilisent dans leurs branches va créer un conflit lorsque vous refusionnez votre branche avec le projet racine.
- Fusionnez les branches dans l'ordre inverse de leur création. Par exemple, Philippe crée une branche à partir du projet racine. Suzanne crée une branche à

partir de la branche de Philippe. Lorsque le moment est venu de refusionner les branches avec le projet racine, Suzanne commence par refusionner sa branche avec la branche de Philippe, puis Philippe refusionne sa branche avec le projet racine.

- Refusionnez souvent les branches avec le projet racine chaque fois que vous apportez quelques modifications à une branche au lieu d'attendre d'avoir beaucoup modifié les branches. Il est ainsi plus facile de résoudre les conflits pendant le processus de fusion. Vous pouvez refusionner toutes les branches avec le projet racine, puis créer à nouveau un branchement pour le projet. Chaque membre de l'équipe reçoit une version mise à jour du modèle.
- Dans un projet branché, toutes les transactions archivées ne seront pas disponibles lorsque vous refusionnez les branches avec le projet principal. Pour plus d'informations, voir «Affichage et enregistrement d'un historique des transactions», à la page 299

## Méthodologies de branchement

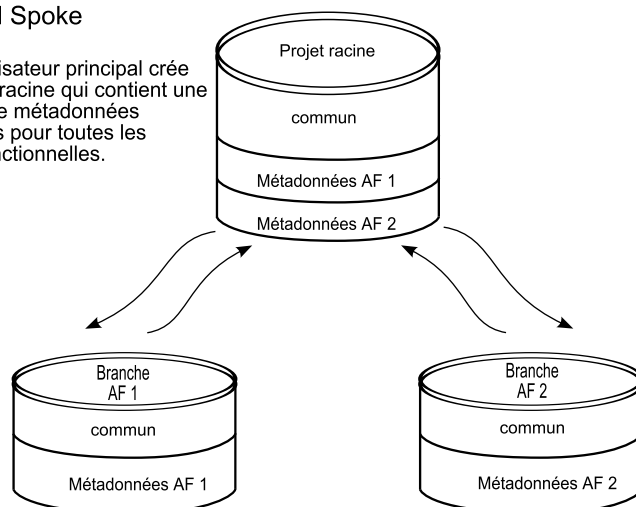
Avant de brancher un projet, vous devez décider du mode de partage des métadonnées dans votre projet. Cette section décrit certaines des méthodologies utilisées couramment pour le partage des métadonnées.

### Hub and Spoke

Avec la méthodologie Hub and Spoke, un projet utilise des métadonnées communes qui doivent être partagées par toutes les zones fonctionnelles. Le projet racine se compose d'une couche physique complètement modélisée contenant les objets requis par toutes les zones fonctionnelles. Le projet racine est branché pour chaque zone fonctionnelle. Chaque zone fonctionnelle peut créer ses propres branches si plusieurs personnes travaillent dessus. A tout moment, un modélisateur de zone fonctionnelle peut refusionner une branche dans le projet racine pour mettre ce dernier à jour, puis effectuer un nouveau branchement pour recevoir les mises à jour. Les objets qui sont communs à toutes les zones fonctionnelles sont conservés dans le projet racine.

#### Hub and Spoke

Le modélisateur principal crée un projet racine qui contient une couche de métadonnées partagées pour toutes les zones fonctionnelles.



Le modélisateur étend cette branche en important des sources de données, en ajoutant des calculs et en publiant des packs spécifiques pour la zone fonctionnelle 1.

Le modélisateur étend cette branche en important des sources de données, en ajoutant des calculs et en créant et publiant des packs spécifiques pour la zone fonctionnelle 2.

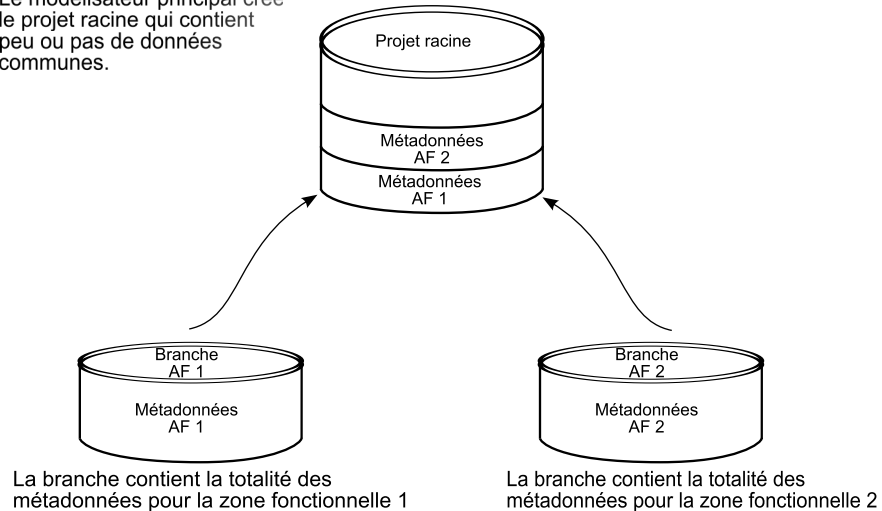


## Métadonnées spécifiques des zones fonctionnelles

Avec cette méthodologie, le projet comporte peu ou pas de métadonnées communes. Chaque zone fonctionnelle développe ses propres objets de façon indépendante dans le projet. Une zone fonctionnelle n'a aucune connaissance des objets présents dans les autres zones fonctionnelles. Le modélisateur principal contrôle la fusion des branches pour empêcher chaque zone fonctionnelle de voir les objets des autres zones fonctionnelles.

### Métadonnées spécifiques des zones fonctionnelles

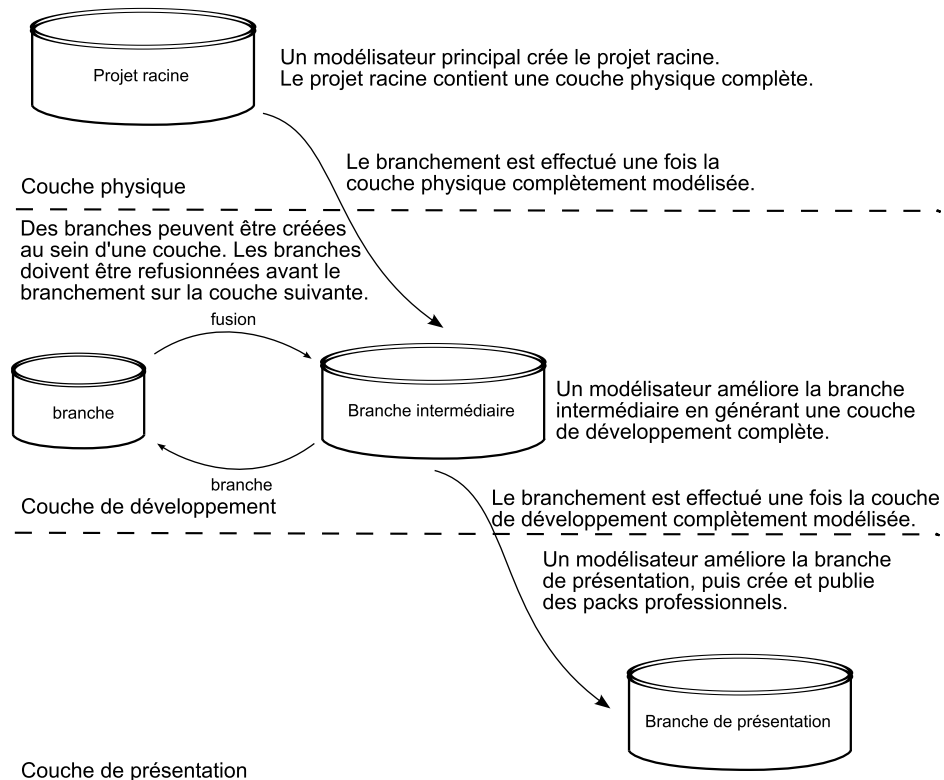
Le modélisateur principal crée le projet racine qui contient peu ou pas de données communes.



## Distribution par couches

Avec la méthodologie de distribution par couches, les métadonnées sont organisées par couches. Chaque couche requiert l'accès aux métadonnées de la couche de niveau supérieur. Par exemple, un modèle trois couches. La couche supérieure correspond au projet racine, composée d'une couche physique complètement modélisée. Le projet racine est modifié pour créer la deuxième couche, la branche intermédiaire. La branche intermédiaire contient une couche de développement complètement modélisée. La branche intermédiaire est branchée pour créer la troisième couche, la branche de présentation. La branche de présentation contient une couche métier complètement modélisée où les packs de génération de rapports sont définis et publiés.

## Distribution par couches



## Méthodes de branchement d'un projet

Voici quelques méthodes permettant de scinder un projet :

- Par ensemble de tâches et de compétences  
Créez des branches distinctes pour les utilisateurs selon qu'ils travaillent sur des schémas en étoile et des packs, sur des calculs et des filtres, sur des dimensions, ou sur des requêtes et des dénominations d'objet.
- Par couches de modèle  
Créez des branches distinctes pour la couche d'importation, la couche intermédiaire et la couche de présentation.
- Par unité commerciale ou domaine fonctionnel  
Créez des branches distinctes pour les ventes et les détaillants.
- Par type de source de données  
Créez des branches distinctes pour pour les sources de données relationnelles, dimensionnelles et ERP.

## Création de branches

Vous pouvez créer une branche dans un projet.

Le fichier journal associé à la nouvelle branche sera vide. Lorsque vous créez une branche pour un projet en lecture seule, la branche résultante est accessible en écriture.

## Procédure

1. Ouvrez le projet à partir duquel vous souhaitez créer une branche.
2. Cliquez sur **Projet, Effectuer un branchement vers**.

3. Dans la zone **Nom du projet**, saisissez un nom pour le projet.
4. Dans la zone **Emplacement**, saisissez le chemin d'accès au dossier dans lequel vous souhaitez stocker la version branchée.

## Fusion de projets

Vous pouvez refusionner une branche avec son projet principal.

Pour refusionner une branche avec le projet principal, IBM Cognos Framework Manager affiche la liste des transactions qui ont été exécutées sur la branche. Une case à cocher figure en regard de chaque transaction. Pour chaque transaction dont la case est cochée, Framework Manager tente d'exécuter la transaction sur le projet avec lequel vous effectuez la fusion. Si une transaction aboutit, elle est signalée par une coche et la transaction suivante est lancée. Si une transaction échoue, elle est signalée par un "X" et le traitement s'arrête.

Lorsqu'une transaction échoue, vous pouvez être en mesure de résoudre immédiatement le problème. Par exemple, si un objet est manquant, Framework Manager identifie l'objet manquant et vous invite à le remplacer par un autre objet.

Lorsque vous atteignez la fin de la liste des transactions, vous pouvez accepter ou annuler les modifications. Si vous acceptez les modifications, les transactions qui ont abouti sont supprimées de la liste des transactions. Si vous n'acceptez pas les modifications, la liste de transactions n'est pas mise à jour et vous pouvez effectuer à nouveau la même fusion ultérieurement.

Certains conflits ne peuvent pas être résolus au cours du processus de fusion, ou plusieurs de transaction peuvent être liées à une transaction ayant échoué. Dans ce cas, vous pouvez ignorer les transactions en cochant la case **Décocher les transactions dépendantes**. Framework Manager désélectionne alors la case à cocher pour chaque transaction liée à la transaction ayant échoué. Lorsque Framework Manager reprend l'exécution de la liste des transactions, il n'exécute pas les transactions dont la case est désélectionnée. Lorsque vous atteignez la fin de la liste des transactions, vous pouvez accepter les modifications jusqu'à ce point, identifier les transactions qui ont été ignorées, corriger les erreurs correspondantes, puis effectuer à nouveau la fusion. Lorsque vous exécutez à nouveau la fusion, la liste des transactions contient uniquement les transactions qui n'ont pas été exécutées.

**Remarque :** Avant l'exécution de la liste des transactions, Framework Manager effectue une sauvegarde du projet de fusion. La sauvegarde est stockée dans le même répertoire que le projet de fusion.

## Procédure

1. Ouvrez le projet avec lequel vous souhaitez fusionner une branche.
2. Cliquez sur **Projet, Fusionner à partir de**.
3. Dans la boîte de dialogue **Sélection d'un projet à partir duquel lancer la fusion**, cliquez sur **Fichiers de type, Tous les fichiers (\*.\*)**.
4. Recherchez le fichier log.xml pour la branche à fusionner, puis cliquez sur **Ouvrir**.

La fenêtre **Exécution de la fusion** s'ouvre avec une liste de transactions. Les transactions que vous avez sélectionnées sont exécutées.

Framework Manager a uniquement besoin des fichiers log.xml et IdLog.xml, et non de tous les fichiers du projet pour alimenter la liste historique des transactions. Si vous n'ouvrez pas directement le fichier .cpf lorsque vous y êtes

invité, Framework Manager recherche et ouvre le fichier log.xml. Ouvrir directement le fichier log.xml a pour avantage de réduire le nombre de fichiers volumineux susceptibles de devoir être distribués dans un environnement multiutilisateur.

5. Sélectionnez le mode de traitement des transactions :
  - Pour exécuter la totalité de la liste des transactions en continu du début à la fin, cliquez sur **Exécuter**.
  - Pour exécuter une transaction, puis vous interrompre, cliquez sur **Etape**.

Lorsqu'une transaction est terminée, une coche ou un "X" s'affiche en regard de celle-ci. Une coche indique que la transaction a été appliquée avec succès au modèle avec lequel vous effectuez la fusion. Un "X" signifie que la transaction a échoué. Des informations détaillées pour chaque transaction sont répertoriées sous **Détails des transactions**. Si une transaction échoue, le processus de fusion s'interrompt.
6. Effectuez l'une des opérations suivantes :

Objectif	Action
Corriger une transaction ayant échoué	Voir «Correction de transactions ayant échoué».
Ignorer la transaction en cours et exécuter la transaction suivante	Cliquez sur <b>Omettre</b> .
Exécuter la transaction en cours et s'interrompre	Cliquez sur <b>Etape</b> .
Exécuter la liste des transactions de la transaction en cours à la fin	Cliquez sur <b>Continuer</b> .
Accepter les transactions exécutées jusqu'à ce point et revenir à la page de projet	Cliquez sur <b>Accepter</b> .
Annuler toutes les transactions exécutées jusqu'à ce point et revenir à la page de projet	Cliquez sur <b>Annuler</b> .

7. Exécutez l'étape précédente jusqu'à ce que vous atteigniez la fin de la liste des transactions ou que vous ayez accepté ou annulé les modifications.
 

Si vous avez accepté les modifications, elles s'affichent dans l'**afficheur de projet**. Si vous avez annulé les modifications, aucune modification ne s'affiche.
8. Si vous avez accepté les modifications, enregistrez le projet fusionné dans la page de projet. Si vous décidez de ne pas enregistrer les modifications, fermez le projet sans l'enregistrer.

**Remarque :** Si vous acceptez les modifications dans la fenêtre **Exécution de la fusion**, mais que vous n'enregistrez pas le projet, vous ne pourrez plus exécuter la même fusion. Une fois que vous avez accepté les transactions, celles-ci sont supprimées de la liste des transactions.

#### Correction de transactions ayant échoué :

Lors du processus de fusion, des transactions peuvent échouer. La plupart des transactions peut être corrigée en remplaçant un objet par un autre. Il est possible de corriger toutes les transactions en modifiant le projet.

### Cas 1

Un objet utilisé dans le projet de branche ne figure pas dans le projet racine. Dans la boîte de dialogue **Fusionner**, la zone **Remplacer** indique le nom de l'objet qui est manquant dans le projet racine.

Pour résoudre ce problème, vous pouvez sélectionner un autre objet dans le projet racine en procédant comme suit :

- Dans l'**afficheur de projet** ou le **diagramme**, sélectionnez le nom de l'objet à utiliser à la place de l'objet manquant.  
L'objet s'affiche dans la **zone de remplacement** de la boîte de dialogue **Fusionner**.
- Cliquez sur **Remplacer**.  
La transaction s'exécute de nouveau, en utilisant votre objet de remplacement pour l'objet manquant.

Pour les problèmes plus complexes, vous pouvez être invité à sélectionner l'objet introuvable dans le projet racine à partir de la boîte de dialogue **Résolution de conflit d'appellation d'objet**. Si un avertissement s'affiche, cela signifie que l'incident ne peut pas être résolu à l'aide de cette boîte de dialogue. En revanche, vous pouvez modifier le projet racine ou ignorer la transaction.

### Cas 2

Pour toutes les transactions ayant échoué, vous pouvez résoudre le problème en modifiant le projet racine. Tout d'abord, modifiez le modèle comme vous le souhaitez, puis, dans la boîte de dialogue **Exécution de la fusion**, cliquez sur **Continuer** pour exécuter l'intégralité de la liste de transactions en commençant par la transaction qui a échoué.

### Cas 3

Vous ne parvenez pas à corriger la transaction. Pour résoudre ce problème, procédez comme suit :

- Cliquez sur **Ignorer** pour ignorer la transaction suivante.
- Désélectionnez la case à cocher correspondant à la transaction ayant échoué.
- Assurez-vous que la case à cocher **Décocher les transactions dépendantes** est sélectionnée.
- Dans la boîte de dialogue **Exécution de la fusion**, cliquez sur **Continuer**.

## Projets segmentés et liés

Vous pouvez utiliser IBM Cognos Framework Manager pour créer et lier des segments, des projets et des dossiers. Un segment est un projet au sein du projet principal. Un segment appartient à son projet principal. Un lien est un raccourci vers un projet existant. Le projet qui est lié peut être partagé par plusieurs projets.

Un segment de projet est un projet complet et les modifications apportées à ce projet ont un impact sur tous les projets qui lui sont liés. Si vous souhaitez ouvrir un segment en tant que projet distinct, celui-ci doit être structuré sous la forme d'un projet complet. Il doit exister une couche physique dans chaque segment contenant un sous-ensemble des sujets de requête de source de données sur

laquelle ils sont basés. Ces sujets de requête de source de données permettent d'accéder aux données et aux métadonnées, et doivent être inclus dans les segments appropriés.

Ne modifiez pas la couche physique d'un segment. Les modifications seraient répercutées dans le modèle parent lié et auraient un impact sur tous les segments du modèle qui partagent des sujets de requête de source de données. Les modifications peuvent ne pas être visibles à l'extérieur du modèle dans lequel elles ont été effectuées tant que le modèle n'a pas été fermé et rouvert.

Avant de segmenter un projet, assurez-vous que le dossier et l'espace-noms sont nommés correctement. Vous ne pourrez pas renommer le dossier ou l'espace-noms une fois que projet segmenté.

Les modifications apportées au modèle racine, telles que la mise à niveau et la configuration de la sécurité des objets, ne sont pas héritées par le modèle segmenté. Vous devez appliquer les modifications à chaque segment du modèle.

Les paramètres de restriction ont la priorité sur les paramètres appliqués au modèle actuellement ouvert (qu'il s'agisse d'un modèle parent ou enfant).

Le projet principal a accès à l'ensemble du modèle, y compris les segments. Vous pouvez apporter des modifications aux segments lorsque vous travaillez dans le projet principal. Cependant, si plusieurs utilisateurs accèdent au segment, il existe un risque potentiel de pertes de mises à jour.

**Remarque :** Lorsque vous modifiez la structure du projet, n'ouvrez pas les segments en tant que projets individuels. Vérifiez plutôt le projet principal et effectuez les modifications à partir de celui-ci.

Vous pouvez lier les segments à d'autres projets qui contiennent des informations liées afin de garantir une cohérence et de réutiliser les informations. Si vous envisagez de lier des segments de modèle, suivez la procédure de segmentation des projets. Il est conseillé d'établir des liens vers des segments relativement complets et de vérifier régulièrement le modèle.

## Segmentation de projets

Une bonne maîtrise de la segmentation des projets est essentielle pour garantir la stabilité et réduire la complexité dans un environnement de modélisation multiutilisateur. Si vous prévoyez de segmenter votre projet, procédez comme suit :

- Modélisez la couche physique aussi complètement que possible :
  - En veillant à ce que l'espace-noms du projet principal et les liens au sein du projet vers les dossiers aient le même identificateur. Voir «Importation d'objets portant le même nom», à la page 65.  
Par exemple, vous disposez d'un projet principal et d'un lien dans le projet vers un dossier. Le dossier que vous liez doit exister dans un espace-noms portant le même nom que le projet principal. Si l'identificateur dans le projet principal et celui du dossier lié ne sont pas les mêmes, les relations, les raccourcis ou les expressions qui font référence à des objets dans le lien depuis le projet principal risquent de ne pas fonctionner.
  - En vérifiant que tous les objets d'un projet ont des identificateurs uniques  
Par exemple, vous disposez d'un projet principal contenant un sujet de requête nommé NewQS et d'un segment dans le projet. Vous ouvrez le

segment, vous ajoutez un nouveau sujet de requête nommé NewQS et vous enregistrez le segment. Lorsque vous ouvrez le projet principal, une erreur se produit car un sujet de requête nommé NewQS existe déjà dans le projet principal.

- En mettant à jour les références dans le projet principal et dans les segments du projet

Par exemple, vous disposez d'un projet principal et d'un segment dans le projet. Dans le projet principal, une relation nommée qs1\_qs2 existe entre sujet\_requête1 et sujet\_requête2. Le sujet de requête sujet\_requête2 est dans le segment. Vous ouvrez le segment, vous renommez sujet\_requête2 en sujet\_requête3 et vous enregistrez le segment. Lorsque vous ouvrez le projet principal, une erreur se produit car la relation qs1\_qs2 est rompue. Dans Framework Manager, tous les objets qui s'appuient sur une référence, tels que les raccourcis, les sujets de requête de modèle et les expressions sont également affectés.

- En veillant à ce que le projet principal et tous les segments du projet soient associés aux mêmes langues

Par exemple, vous disposez d'un projet principal et d'un segment dans le projet. Dans le segment, vous avez défini l'Anglais et le Français. Vous ouvrez le projet principal, vous ajoutez le Chinois et vous enregistrez le segment. Lorsque vous ouvrez le segment, une erreur se produit car la langue Chinois n'est pas définie dans le segment.

- Organisez la couche physique à l'aide d'espaces-noms.

Vous devez créer un espace-noms pour les sujets de requête, les calculs et les filtres dont vous pensez qu'ils seront nécessaires pour plusieurs segments.

Vous devez créer un espace-noms pour chaque collection de sujets de requête qui est unique pour un segment de modèle prévu.

- Acceptez le nom de projet par défaut lors de la création du projet segmenté.

Le projet segmenté doit être créé dans un sous-dossier du dossier du projet principal. Le nom de projet par défaut est le même que celui du dossier ou de l'espace-noms qui le contient.

- Segmentez le modèle pour chaque espace-noms que vous avez créé.

- Utilisez dans la mesure du possible un référentiel de contrôle des sources pour restreindre l'accès et effectuer un suivi des modifications de vos projets et segments.

## **Limites relatives à la segmentation et à la liaison de projets**

Les limites qui s'appliquent à la segmentation et à la liaison de projets sont les suivantes :

- Vous ne pouvez pas tester les objets d'un segment ou d'un projet lié qui font référence à des objets figurant dans un segment qui n'est pas disponible.
- Vous ne pouvez pas créer de nouveaux objets dans un segment ou un projet lié qui font référence à des objets figurant dans un segment qui n'est pas disponible.
- Lorsque vous établissez une liaison vers un projet, tous les objets référencés (objets d'espace-noms, sources de données et tables de mappage des paramètres) sont liés. Les packs ne sont pas liés.
- Les modifications que vous apportez à un segment enfant ne sont pas répercutées dans le projet principal, même après une actualisation de l'écran à l'aide de la touche F5. Cela se produit car un autre segment enfant lié au parent ou au projet principal lui-même est ouvert. Fermez tous les projets de segment enfant, puis rouvrez le projet principal.

- L'endroit où le segment est visible dépend de l'endroit où vous le créez dans un projet. Si vous créez un segment imbriqué à partir du projet principal, ce segment imbriqué est visible à partir du projet principal. Si vous ouvrez le segment contenant le segment imbriqué sous la forme d'un projet autonome, le segment imbriqué n'est pas visible. Si vous créez un segment imbriqué à partir d'un segment ouvert en tant que projet autonome, ce segment imbriqué est visible à partir du projet autonome. Si vous ouvrez le projet principal, le segment imbriqué créé à partir du projet autonome n'est pas visible.

## Création de segments

Avec des segments, vous pouvez organiser un projet en fonction de règles métier et d'exigences organisationnelles, et partager et réutiliser les informations d'un projet.

Vous créez des segments au niveau des dossiers ou des espaces-noms. Vous pouvez créer un nouveau projet dans un nouveau dossier, ou le compléter avec ses propres fichiers de projet.

Lorsqu'un segment est créé, les tables de mappage des paramètres existantes du projet principal sont copiées dans le nouveau segment. Une fois le segment créé, les tables de mappage des paramètres sont uniques pour chaque segment et ne peuvent pas être partagées entre les segments. Par exemple, si vous travaillez dans le projet principal, vous pouvez utiliser une table de mappage de paramètres dans un sujet de requête appartenant à un segment. Par contre, si vous ouvrez le segment, la table de mappage des paramètres n'est pas disponible.

Vous ne pouvez accéder à un segment qu'à partir du projet dans lequel il a été créé. Par exemple, vous ouvrez le projet principal et vous créez un segment (Segment A). Vous créez ensuite un autre segment (Segment B) dans Segment A. A partir du projet principal, vous pouvez voir Segment A et Segment B. Cependant, si vous ouvrez Segment A tout seul, vous ne voyez pas Segment B.

Avant de créer des segments, envisagez de scinder votre projet en unités commerciales. Par exemple, vous disposez d'un projet nommé Ventes. Vous pouvez créer deux dossiers, l'un nommé Produits et l'autre, Commandes. Vous pouvez scinder le projet Ventes au niveau du dossier Produits et du dossier Commandes.

## Procédure

1. Cliquez sur le dossier ou l'espace-noms à diviser.
2. Cliquez sur **Projet, Créer un segment**.  
Vous pouvez accepter le paramètre par défaut pour le nom du projet.
3. Pour renommer le segment, dans la zone **Nom du projet**, saisissez un autre nom.  
Cela ne modifie pas le nom du dossier. Si vous souhaitez renommer le dossier, vous devez le renommer dans l'**afficheur de projet** avant de créer le segment.  
Pour faciliter l'utilisation, conservez le même nom pour le dossier et le segment.
4. Cliquez sur **OK**.  
L'**afficheur de projet** est actualisé et les icônes représentant le dossier segmenté ou l'espace-noms segmenté sont affichées.

## Création de liens

Vous pouvez créer des liens pour mieux organiser le travail sur des grands projets, assurer une cohérence, et réutiliser des informations.



Par exemple, le projet nommé Stock contient le dossier nommé Produits. Vous pouvez créer un lien depuis Produits en vente vers Produits en stock. Si des modifications ou des ajouts sont effectués dans le dossier Produits en stock, vous les verrez dans le dossier Produits en vente.

Si vous envisagez de lier des segments de modèle, vous devez suivre la procédure de segmentation de modèle. Pour plus d'informations, voir «Segmentation de projets», à la page 284.

Un projet lié est partagé par d'autres projets. Il ne doit pas être créé dans un sous-répertoire du répertoire du projet principal.

### **Avant de commencer**

Vous devez créer le projet, le dossier ou l'espace-noms pour pouvoir établir un lien vers celui-ci.

La même langue de conception et les mêmes langues doivent être définies pour les projets que vous liez.

### **Procédure**

1. Dans l'**afficheur de projet**, cliquez sur le projet, le segment, l'espace-noms ou le dossier vers lequel établir le lien.

**Astuce :** Vous ne pouvez créer des liens que vers des dossiers, des espaces-noms, des projets ou des segments.

2. Cliquez sur **Projet, Lier le segment**.
3. Recherchez et cliquez sur le fichier .cpf du projet contenant l'objet vers lequel établir le lien.
4. Cliquez sur **Ouvrir**.
  - Si le projet que vous avez sélectionné nécessite une mise à niveau, vous êtes invité à l'effectuer. Pour plus d'informations, voir Chapitre 11, «Mise à niveau de modèles», à la page 383.
  - Si le projet utilise un identificateur d'unité mappé, vous êtes invité à conserver cet identificateur ou à le changer en chemin UNC.  
Vous devez opter pour le chemin UNC si votre projet ne sera partagé par d'autres projets.
5. Sélectionnez le projet, le segment, l'espace-noms ou le dossier vers lequel établir le lien :
  - Pour établir un lien vers un autre projet, cliquez sur **Ajouter un projet**, recherchez le fichier .cpf et cliquez sur **Ouvrir**. Sélectionnez le projet et cliquez sur **Ajouter**.
  - Pour établir un lien vers un segment, cliquez sur le segment, puis sur **Ajouter**.
6. Cliquez sur **OK**.  
Un nouveau dossier s'affiche dans l'**afficheur de projet**.

### **Tirer parti des projets en lecture seule**

Vous pouvez rendre un projet en lecture seule disponible pour d'autres développeurs tout en le protégeant contre les modifications indésirables.

Vous pouvez mettre un projet en lecture seule en modifiant les propriétés du fichier. L'interface utilisateur est alors modifiée. Les noms, les propriétés et les actions grisés indiquent qu'aucune modification ne peut être effectuée.

## Procédure

1. Créez un partage pour héberger le projet qui doit être protégé.
2. Accordez un accès en lecture seule à ce partage pour que les développeurs puissent tirer parti du projet.

## Contrôle de référentiel externe

Vous pouvez utiliser IBM Cognos Framework Manager avec un système de contrôle de source externe. Cette procédure explique comment placer les fichiers de projet Framework Manager dans un référentiel externe. Vous pouvez travailler sur le projet dans Framework Manager et le référentiel externe peut gérer le contrôle des versions des fichiers du projet.

Pour utiliser un système de contrôle de source externe, procédez comme suit :

- Assurez-vous que le fichier est fermé.
- Supprimez le fichier repository.xml s'il existe.
- Ajoutez au référentiel externe tous les fichiers qui existent dans le répertoire du projet.

Le répertoire du projet est le répertoire qui contient le fichier *nom projet.cpf*.

- Extrayez les fichiers du projet du référentiel externe.
- Travaillez sur le projet dans Framework Manager.
- Enregistrez les modifications.
- Restituez les fichiers du projet dans le référentiel externe.

## Segmentation de projets

Vous pouvez créer un segment pour un projet stocké dans un référentiel externe. Les segments sont des répertoires de projet stockés sous le répertoire de projet principal. Conservez la même hiérarchie dans le référentiel que dans le répertoire de projet.

Les segments peuvent être ouverts individuellement en tant que projets autonome.

Un segment peut également être ouvert dans le cadre du projet principal. Dans ce cas, vous devez extraire le projet pour chaque segment que vous souhaitez modifier.

---

## Administration des métadonnées

Vous pouvez modifier les métadonnées dans vos modèles pour répondre à vos objectifs de modélisation.

Vous pouvez effectuer les opérations suivantes :

- Copier, déplacer, renommer ou supprimer des projets pour les organiser de façon pertinente pour votre environnement de génération de rapports.
- Analyser les modifications apportées à un modèle pour voir leur impact sur les packs et les rapports qui utilisent ce modèle.
- Remapper un objet à une nouvelle source.
- Exporter votre modèle pour échanger des métadonnées entre les différents outils d'entrepôt de données et référentiels.
- Réutiliser les mêmes modèles et rapports avec différents ensembles de données.
- Déplacer un modèle d'une base de données relationnelle vers une autre.

## Copie, déplacement, changement de nom ou suppression de projets

Il est conseillé d'organiser les projets de façon significative afin de pouvoir les retrouver facilement. Dans IBM Cognos Framework Manager, vous pouvez copier, déplacer, renommer et supprimer des projets.

Vous pouvez gérer vos projets à l'aide des fonctions de segmentation et de liaison. Ces fonctions de gestion de projet permettent d'organiser un projet en fonction de règles métier et de besoins organisationnels, de définir des options de traitement et d'accorder à d'autres utilisateurs les droits nécessaires pour accéder à des sections du projet. Pour plus d'informations, voir «Création de segments», à la page 286 et «Création de liens», à la page 286.

Vous pouvez également identifier les fonctions de bases de données que vous souhaitez utiliser pour chaque source de données que vous importez dans votre projet. Pour plus d'informations, voir «Sélection d'ensembles de fonctions», à la page 321

Si votre projet est segmenté, les segments sont traités comme des projets autonomes. Si vous sauvegardez ou copiez un projet dans un projet existant, il est traité comme un segment.

### Copie de projets

Lorsque vous copiez un projet, vous en créez une réplique à un autre emplacement. Tous les fichiers du dossier du projet, y compris les sous-dossiers, sont copiés dans le nouvel emplacement. Lorsque vous apportez des modifications au projet dans un dossier, ces modifications ne sont pas répercutées dans les copies du projet dans d'autres dossiers.

La copie d'un modèle segmenté copie tous les segments, ainsi que le projet principal.

Il se peut parfois que vous ne puissiez pas copier un projet et que vous deviez plutôt utiliser **Enregistrer sous**. L'enregistrement d'un projet sous un nouveau nom crée un nouveau dossier de projet tandis que l'enregistrement du projet sous le nom existant écrase le projet en cours. Cela s'avère utile si vous souhaitez enregistrer des modifications apportées à un projet en lecture seule, ou si vous souhaitez enregistrer un projet sous un autre nom ou dans un nouvel emplacement sans écraser le projet d'origine.

Vous ne pouvez pas créer une copie d'un projet dans le même dossier que le dossier d'origine. Si vous copiez un projet sous un dossier de projet existant, Framework Manager le traite en tant que segment de projet. Pour plus d'informations, voir «Projets segmentés et liés», à la page 283.

Si un projet ou un segment est ouvert lorsque vous le copiez, la dernière version enregistrée est copiée.

### Procédure

1. Cliquez sur **Fichier, Gérer les projets, Copier**.
2. Dans la zone **De**, cliquez sur le bouton Parcourir et sélectionnez le fichier .cpf pour le projet que vous souhaitez copier.  
**Remarque** : Le nom du dossier du projet est affiché dans la zone de texte.
3. Dans la zone **Vers**, entrez le nom du projet.

Par défaut, le nom du projet et le répertoire dans lequel le projet est enregistré sont les mêmes.

4. Dans la zone **Emplacement**, indiquez le nouvel emplacement ou cliquez sur le bouton Parcourir et sélectionnez le nouvel emplacement du projet.
5. Cliquez sur **OK**.

## Déplacement de projets

Vous pouvez décider de déplacer un projet si votre dossier se remplit au point qu'il devient difficile de retrouver un projet particulier. Lorsque vous déplacez un projet, vous le copiez en fait dans un nouveau dossier et vous le supprimez du dossier en cours. Tous les fichiers du dossier du projet, y compris les sous-dossiers, sont déplacés vers le nouvel emplacement.

Le déplacement d'un modèle segmenté déplace tous les segments, ainsi que le projet principal.

Pour que vous puissiez déplacer un projet, le projet doit être fermé dans Framework Manager.

### Procédure

1. Cliquez sur **Fichier, Gérer les projets, Déplacer**.
2. Dans la zone **De**, cliquez sur le bouton Parcourir et sélectionnez le fichier .cpf pour le projet que vous souhaitez déplacer.  
**Remarque** : Le nom du dossier du projet est affiché dans la zone de texte.
3. Dans la zone **Vers**, indiquez le nouvel emplacement ou cliquez sur le bouton Parcourir et sélectionnez le nouvel emplacement du projet.
4. Cliquez sur **OK**.

## Changement nom pour des projets

Lorsque vous renommez un projet, vous fournissez un nouveau nom pour le fichier .cpf. Vous ne modifiez pas l'emplacement du projet. Les fichiers de projet secondaires et les fichiers journaux conservent leur nom d'origine.

Si un projet s'affiche dans la liste des projets récents dans la page **Bienvenue** de Framework Manager et que vous le renommez, vous ne pourrez pas ouvrir le projet en cliquant sur le lien. Vous devez ouvrir le projet à l'aide de la commande **Ouvrir** du menu **Fichier**.

Pour que vous puissiez renommer un projet, le projet doit être fermé dans Framework Manager.

### Procédure

1. Cliquez sur **Fichier, Gérer les projets, Renommer**.
2. Dans la zone **De**, cliquez sur le bouton Parcourir et sélectionnez le fichier .cpf pour le projet que vous souhaitez renommer.  
**Remarque** : Le nom du dossier du projet est affiché dans la zone de texte.
3. Dans la zone **Vers**, saisissez le nouveau nom pour le projet, puis cliquez sur **OK**.

Si le dossier du projet d'origine et le fichier .cpf ont le même nom, le dossier et le fichier .cpf sont renommés.

## Suppression de projets

Lorsque vous supprimez un projet, le dossier du projet et tout son contenu, y compris les fichiers utilisateur, sont supprimés du système de fichiers et envoyés dans la corbeille de vidage.

Si votre projet est segmentée et que vous supprimez le projet principal, les segments sont également supprimés. La suppression d'un segment de projet supprime uniquement le segment, pas le modèle sur lequel il est basé.

En règle générale, il est conseillé de supprimer les segments à partir du modèle. Si vous supprimez le segment en utilisant **Supprimer** dans le menu **Fichier**, le segment s'affiche comme s'il existait toujours dans le modèle. Pour plus d'informations, voir «Projets segmentés et liés», à la page 283.

Avant de supprimer un projet, assurez-vous que le projet et tous ses segments sont fermés. Framework Manager ne prend pas en charge de mécanisme de verrouillage de fichier. Il est donc possible dans certains cas de supprimer un projet avec des segments ouverts. Si vous supprimez un projet avec des segments ouverts, les segments ne pourront plus être enregistrés.

### Procédure

1. Cliquez sur **Fichier, Gérer, Projets, Supprimer**.
2. Dans la zone **Dossier du projet**, cliquez sur le bouton Parcourir et sélectionnez le fichier .cpf pour le projet que vous souhaitez supprimer.

**Remarque :** Le nom du dossier du projet est affiché dans la zone de texte.

3. Cliquez sur **OK**.

Le dossier du projet et tout son contenu sont supprimés.

## Analyse de l'impact des modifications apportées à des packs

Avant de publier un pack et d'exécuter des rapports, vous pouvez voir comment les modifications que vous apportez à un modèle vont affecter le pack et les rapports qui l'utilisent. Vous pouvez rechercher les modifications qui ont été effectuées sur le pack et afficher les détails de chaque modification ainsi que son incidence sur les rapports.

Les rapports qui sont créés à l'aide du pack peuvent être affectés par les modifications que vous avez apportées au modèle. Par exemple, l'ajout de nouveaux objets à un pack n'a pas d'incidence sur un rapport, mais le changement de nom d'un élément de requête en a une. La définition de rapport ne sera pas valide car l'élément de requête ne figurera pas dans la définition du pack. Si vous utilisez la fonction de modèle durable, vous pouvez éviter l'impact des changements de nom d'élément de requête sur les rapports. Pour plus d'informations, voir «Modèles durables», à la page 187.

**Remarque :** Comme un rapport utilise un pack publié, si vous apportez des modifications au modèle, mais que vous ne publiez pas le pack qui l'utilise, le rapport n'est pas affecté par les modifications.

Si vous changez le nom d'un objet, il s'affiche avec la mention "modified" dans les résultats de l'analyse.

L'analyse est effectuée sur les objets utilisés directement par un modèle, ainsi que sur les objets sous-jacents. Par exemple, vous disposez d'un sujet de requête de modèle qui est basé sur un sujet de requête de source de données. Si vous

modifiez le sujet de requête de modèle, il apparaîtra en tant qu'objet modifié. Si vous modifiez le sujet de requête de source de données, il s'affichera également en tant qu'objet modifié.

Les types d'objets suivants sont analysés : les sujets de requête, les éléments de requête, les mesures, des dimensions ordinaires, les dimensions de mesure, les hiérarchies, les niveaux, les filtres autonomes et les calculs autonomes.

## Procédure

1. Dans l'**afficheur de projet**, cliquez sur un pack qui a été publié.
2. Cliquez sur **Actions, Pack, Analyser l'impact de publication**.
3. Sélectionnez l'action à exécuter :

Objectif	Action
Afficher les dépendances d'un rapport	Voir «Recherche de dépendances de rapport».
Afficher les dépendances d'un objet	Voir «Affichage de dépendances d'objet», à la page 293.
Afficher les détails d'un objet	Cliquez sur la ligne qui contient l'objet.  Les détails concernant l'objet s'affichent sous <b>Détails des modifications pour</b> .
Rechercher un objet dans l' <b>afficheur de projet</b>	Dans la ligne contenant l'objet, sous <b>Actions</b> , cliquez sur <b>Rechercher dans la vue de projet</b> .
Trier les résultats	Cliquez sur <b>Trier</b> en haut de la colonne.
Afficher les commentaires du modélisateur, la personne qui a effectuée la dernière modification et la date de la dernière modification	Cliquez sur la flèche double vers le bas.

4. Cliquez sur **Fermer**.

## Recherche de dépendances de rapport

Vous pouvez rechercher les rapports qui utilisent un objet.

### Procédure

1. Dans la boîte de dialogue **Analyser l'impact de publication**, effectuez l'une des opérations suivantes :
  - Sélectionnez chaque objet pour lequel vous souhaitez identifier les dépendances de rapport en cochant les différentes cases.
  - Sélectionnez tous les objets en cochant la case située en haut de la colonne de cases à cocher.
2. Cliquez sur **Rechercher des dépendances de rapport**.
3. Indiquez la portée de la recherche :

Objectif	Action
Rechercher dans tous les dossiers	Cliquez sur <b>Tous les dossiers</b> .
Limiter la recherche à un dossier spécifique	Cliquez sur <b>Recherche restreinte (parcourir et sélectionner un dossier)</b> . Entrez le nom du dossier ou cliquez sur <b>Parcourir</b> pour rechercher un dossier.

4. Cliquez sur **Rechercher**.  
Une liste de noms de rapport s'affiche dans la fenêtre **Dépendance de rapport** sous **Rapports impactés**. Les résultats affichent les dépendances directes et indirectes. Les noms des objets qui ont un impact indirect sur les rapports sont affichés dans une couleur plus claire.
5. Pour trier les résultats, cliquez sur **Trier** en haut de la colonne.
6. Cliquez sur **Fermer**.

### Affichage de dépendances d'objet

Vous pouvez rechercher des objets qui dépendent d'autres objets ou afficher les dépendances d'un objet enfant.

Vous ne pouvez pas afficher les dépendances pour les tables de mappage des paramètres.

#### Procédure

1. Dans l'**afficheur de projet**, cliquez sur un objet.
2. Cliquez sur **Outils, Afficher les dépendances d'objet**.  
Les objets qui dépendent de l'objet sélectionné s'affichent sous **Objets dépendants**.
3. Pour afficher l'identificateur d'objet pour les objets dépendants, cochez la case **Afficher l'ID objet**.
4. Si l'objet a des enfants et que vous souhaitez afficher les dépendances pour un objet enfant, cliquez sur le signe plus (+) en regard de l'objet contenant l'objet enfant.
5. Cliquez sur un objet enfant sous l'objet parent.  
Les objets qui dépendent de l'objet enfant s'affichent sous **Objets dépendants**.
6. Facultatif : Vous pouvez également afficher des dépendances d'objet de la manière suivante :
  - Dans l'**afficheur de projet**, cliquez avec le bouton droit de la souris sur un objet et sélectionnez **Afficher les dépendances d'objet**.
  - Dans la fenêtre **Context Explorer**, cliquez avec le bouton droit sur un objet et sélectionnez **Afficher les dépendances d'objet**.
  - Dans la fenêtre **Analyser l'impact de publication**, cliquez sur l'icône **Afficher les dépendances** sous **Actions** dans la ligne qui contient l'objet.

## Remappage d'objets à de nouvelles sources

Au cours du cycle de vie d'un modèle IBM Cognos Framework Manager, vous pouvez être amené à modifier la source de données utilisée. Par exemple, vous pouvez souhaiter utiliser le modèle sur une autre base de données avec les mêmes données, faire migrer le modèle d'un schéma transactionnel vers un schéma en étoile ou en flocon, ou encore remplacer une base de données ou une vue d'importation existante par une nouvelle vue. Toutes ces actions peuvent affecter vos rapports. Par exemple, si vous changez les noms d'objets, des rapports peuvent ne plus être validés.

Vous pouvez minimiser l'effet des changements apportés au modèle et à la source de données en remappant des objets du modèle de niveau supérieur afin qu'ils puissent continuer à s'exécuter normalement et à renvoyer des données correctes. Lorsque vous effectuez un remappage, vous pouvez faire correspondre et remplacer des références d'objet ou des noms d'un objet d'origine par des

références ou des noms d'objet d'un autre objet. Vous pouvez remapper des éléments de requête et des mesures. Vous pouvez remapper manuellement des objets individuels ou vous pouvez remapper plusieurs objets en même temps. Lorsque vous remappez plusieurs objets, Framework Manager fait correspondre des éléments de l'objet d'origine avec des éléments de l'autre objet à l'aide des critères de correspondance que vous spécifiez. Seuls les objets qui correspondent aux critères de correspondance sont remappés. Vous pouvez utiliser le nom de l'objet ou la référence d'objet comme critère de correspondance pour les objets d'origine et les autres objets.

Lorsque vous modifiez le critère de correspondance pour le remappage, vous spécifiez les critères qui seront utilisés pour le remappage à des objets ultérieurs.

Si vous remappez un sujet de requête de modèle ou une dimension de modèle qui contient un filtre ou un calcul, le filtre ou le calcul du modèle est également remappé. Aucun message ou avertissement ne s'affiche à ce sujet.

Validez tous les rapports affectés à chaque fois que vous modifiez votre modèle. Pour identifier les rapports affectés, voir «Affichage de dépendances d'objet», à la page 293 et «Recherche de dépendances de rapport», à la page 292

Vous ne pouvez pas remapper des sujets de requête de source de données ou des dimensions de source de données. Le remappage est uniquement pris en charge lors de l'utilisation des paramètres régionaux de conception du modèle.

## Procédure

1. Dans l'**afficheur de projet**, cliquez avec le bouton droit de la souris sur un objet et sélectionnez **Remapper vers la nouvelle source**.
2. Si vous souhaitez modifier les critères de correspondance, cliquez sur **Options** et procédez comme suit :
  - Sélectionnez les critères de correspondance pour l'objet que vous utilisez pour le remappage et pour l'objet d'origine que vous remappez.  
Vous pouvez faire correspondre des objets par nom ou par référence d'objet.
  - Les critères par défaut sont **Par nom** pour l'objet que vous utilisez pour le remappage et **Par référence d'objet** pour l'objet d'origine que vous remappez.
  - Cliquez sur **OK**.
  - Pour utiliser les critères que vous avez spécifiés, cochez la case **Utiliser les options de critères correspondants**.  
Si le critère de correspondance est **Par nom** à **Par nom**, les espaces dans la chaîne sont supprimés.  
S'il n'existe aucune référence d'objet, le nom d'objet est utilisé.
3. Effectuez une ou plusieurs des opérations suivantes :

Objectif	Action
Remapper manuellement un objet individuel	<p>Sous <b>Objets de modèle disponibles</b>, faites glisser un objet vers l'objet que vous souhaitez remapper sous <b>Éléments de requête, mesures, calculs et filtres</b>.</p> <p>La nouvelle valeur pour l'objet s'affiche sous <b>Remapper vers</b>.</p>



Objectif	Action
Remapper automatiquement plusieurs objets	Sous <b>Objets de modèle disponibles</b> , faites glisser un sujet de requête vers n'importe quelle ligne sous <b>Eléments de requête, mesures, calculs et filtres</b> .  Tous les objets qui répondent aux critères de correspondance sont remappés et leurs valeurs s'affichent sous <b>Remapper vers</b> .
Modifier l'expression pour un objet	Cliquez sur les points de suspension (...) en regard de l'objet. Pour en savoir davantage sur la façon de créer une expression, reportez-vous à la section «Création de calculs», à la page 152.
Restaurer une valeur de remappage à la valeur source d'origine	Cliquez avec le bouton droit de la souris sur la ligne contenant l'objet à restaurer et sélectionnez <b>Restaurer sur la valeur initiale</b> .
Effacer la valeur de remappage et la valeur d'origine pour l'objet sélectionné	Cliquez sur la ligne qui contient l'objet, puis sur <b>Effacer</b> .
Effacer la valeur de remappage pour tous les objets	Cliquez sur <b>Effacer tout</b> .

4. Cliquez sur **OK** lorsque vous avez terminé le remappage.

## Exportation de métadonnées

Vous pouvez également exporter votre modèle IBM Cognos Framework Manager sous la forme d'un fichier CWM (Common Warehouse Metamodel). CWM permet d'échanger des métadonnées entre les différents outils d'entrepôt de données et référentiels. Chaque instance du métamodèle CWM est échangée à l'aide de documents XMI (XML Metadata Interchange).

Lorsque vous exportez un modèle Framework Manager sous la forme d'un fichier CWM, les jointures, les dossiers, les espaces-noms, les invites et les calculs ne sont pas exportés. Seuls les sujets de requête, les éléments de requête et les fonctions sont exportés.

Lors d'une exportation vers un fichier CWM, utilisez les options par défaut qui optimisent l'exportation des métadonnées. Modifiez ces options uniquement si vous avez des informations spécifiques concernant l'exportation. Pour plus d'informations sur les options d'exportation, reportez-vous au site Web Meta Integration.

N'utilisez pas des caractères japonais dans le chemin d'exportation.

### Procédure

1. Cliquez avec le bouton droit de la souris sur l'espace-noms racine des métadonnées à exporter, puis cliquez sur **Exporter le modèle**.  
Vous êtes invité à enregistrer le projet.
2. Sélectionnez la cible de l'exportation.
3. Dans la boîte de dialogue **Options d'exportation spécifiques à Framework Manager**, cliquez sur les options souhaitées.

**Remarque :** En règle générale, utilisez les options par défaut. Ces options permettent d'optimiser l'importation des métadonnées. Si vous modifiez les

options, des résultats inattendus risquent de se produire. Pour rétablir les options par défaut, cliquez sur **Utiliser les valeurs par défaut**.

4. Cliquez sur **Suivant**.
5. Dans la boîte de dialogue **Options d'exportation spécifiques à un élément tiers**, utilisez l'option **Fichier** pour identifier le fichier qui doit contenir les métadonnées exportées.

Cliquez sur les autres options souhaitées.

Dans la sous-fenêtre **Description d'option**, vous voyez une description des options disponibles. Les options varient en fonction de la source de données sélectionnée. Pour plus d'informations, voir la documentation du fournisseur sur la source de données.

6. Cliquez sur **Suivant**.  
Les résultats de la validation d'entrée de l'exportation s'affichent.
7. Cliquez sur **Suivant, Terminer**.

## Réutilisation d'un projet

Vous pouvez être amené à utiliser les mêmes modèles et rapports avec différents ensembles de données réelles. Les ensembles de données correspondent à différents comptes, bases de données ou schémas dans une base de données unique.

Vous pouvez être confronté à plusieurs ensembles de données dans les situations suivantes :

- Lorsque vous utilisez un ensemble de données différent de celui utilisé en production.
- Dans les grandes entreprises, où chaque division dispose de son propre ensemble de données.
- Dans les applications OEM qui n'ont pas de contrôle direct sur les données client.

Les tables et les colonnes utilisées par le projet doivent être identiques d'un point de vue logique dans tous les ensembles de données. Vous devez également vous assurer que l'ensemble de données correct est identifié dans chaque cas.

Les sources de données dans IBM Cognos Framework Manager contiennent des informations qui identifient l'emplacement des tables de sources de données requises pour les sujets de requête. Ces informations correspondent au nom de la source de données dans le magasin de contenu, ainsi qu'aux noms de catalogue et de schéma facultatifs. Assurez-vous que les noms de catalogue et de schéma utilisent l'ensemble de données souhaité.

Si des magasins de contenu différents sont utilisés et qu'une version différente du projet est déployée dans chacune d'entre eux, vous pouvez indiquer les informations de source de données du projet pour chaque site. Si vous ne disposez que d'un seul magasin de contenu, vous pouvez publier chaque projet en tant que pack distinct. Ces solutions exigent beaucoup de maintenance manuelle. Pour réduire ce niveau de maintenance, vous pouvez utiliser l'une des options suivantes.

### Identification des informations de source de données

La solution la plus simple consiste à déterminer le nom de la source de données dans le magasin de contenu, le nom du catalogue, le cas échéant, et le nom du schéma dans la base de données. Vous pouvez ensuite utiliser ces noms dans tous les ensembles de données.

Si certains ensembles de données utilisent le même magasin de contenu, créez une connexion distincte pour chaque ensemble de données dans un magasin de contenu unique. Pour en savoir davantage, reportez-vous au *guide d'administration et de sécurité* d'IBM Cognos. Pour plus d'informations sur la manière dont Framework Manager traite plusieurs connexions, voir «Connexions de source de données multiples», à la page 40.

Comme le nom de la source de données du magasin de contenu peut être différent de celui de la base de données client, cette solution offre beaucoup de flexibilité. Toutefois, les noms de catalogue et de schéma doivent être quand-même identiques dans tous les ensembles de données. Même si tous les ensembles de données utilisent le même type de base de données, cela peut-être difficile de la garantir. Si des types de données différents sont impliqués, cela peut s'avérer impossible. Par exemple, SQL Server dispose d'un niveau de catalogue, mais pas Oracle.

### **Utilisation de l'identification d'ensembles de données par défaut utilisateur**

Chaque utilisateur de base de données a accès à un schéma et un catalogue par défaut, le cas échéant. Si le schéma et le catalogue ne sont pas définis, ou s'ils n'ont pas de valeur dans la source de données de projet, la valeur par défaut est utilisée. Comme dans la solution précédente, cette option peut être associée à plusieurs connexions de sorte que des utilisateurs différents puissent utiliser des bases de données différentes pour la même source de données.

Toutefois, lorsque vous modifiez un sujet de requête, IBM Cognos Framework Manager utilise les noms de catalogue et de schéma des sources de données pour les faire correspondre avec les éléments que l'on fait glisser vers les fenêtres SQL à partir de l'arborescence de sources de données. De ce fait, les noms de catalogue et de schéma ne peuvent pas être vides dans la source de données de projet lorsque vous effectuez une modélisation.

Par conséquent, vous devez utiliser une expression de macro dans le catalogue et le schéma de chaque source de données du projet. Cela permet de garantir que les noms de catalogue ou de schéma sont vides au moment de l'exécution, tout en définissant explicitement le catalogue ou le schéma souhaité pendant la modélisation.

### **Procédure de définition explicite d'un catalogue ou d'un schéma**

#### **Procédure**

1. Créez un paramètre de session unique dont la valeur indique si vous êtes en mode conception. Lorsque vous êtes en mode conception, définissez la valeur de ce paramètre de session sur une valeur spécifique, par exemple, conception. Sinon, laissez cette valeur vide.

**Astuce :** Si vous utilisez une valeur de projet ou de remplacement, vous devez la définir chaque fois que vous ouvrez le modèle pour édition.

2. Pour chaque catalogue et schéma de chaque source de données de projet, créez une table de mappage des paramètres qui contient les éléments suivants :
  - Une valeur par défaut vide.
  - Une clé dont le nom correspond à la valeur de conception du paramètre de session ci-dessus et dont la valeur est le nom du catalogue ou du schéma de mode conception pour cette source de données.

3. Sélectionnez la source de données et remplacez les valeurs de propriété de catalogue et de schéma par une macro qui utilise la table de mappage des paramètres et le paramètre de session correspondants.

Par exemple, utilisez :

```
#$DBSchemaName ($DeployOrDesign) #
```

## Portabilité des modèles

Vous pouvez utiliser un modèle IBM Cognos Framework Manager pour accéder à des données issues de différentes instances de base de données. Les instances de base de données peuvent provenir du même fournisseur ou de fournisseurs différents.

Vous devez prendre en compte plusieurs points lorsque vous déplacez un modèle Framework Manager d'une base de données relationnelle vers une autre. Contrairement à ce qui se passe avec le passage d'une base de données à une autre base de données identique sur la même plate-forme, une modification des informations de connexion de source de données peut ne pas suffire.

Vérifiez la génération des déterminants et des relations basées sur des index et ne partez pas du principe que les index décrivent de manière fiable les dépendances ou les relations fonctionnelles pour la génération de rapports.

Les fonctions scalaires sont importées dans un modèle préfixé par une qualification de catalogue ou de schéma dans l'instruction SQL. Comme avec les tables et les vues, vous devrez peut-être supprimer ou modifier la qualification d'emplacement lors d'un changement de fournisseur. Par exemple, si vous créez un modèle sur une base de données ORACLE et que la connexion est modifiée pour pointer vers une base de données SQL Server équivalente, une erreur est générée car le type de source de données de modèle est resté OR et n'a pas été changé en type de source de données approprié.

Pour déplacer un modèle d'une base de données relationnelle vers une autre, procédez comme suit :

- Évaluez le langage de définition de données (DDL) pour déterminer la portabilité pour des noms physiques comme suit :
  - Limitez les noms physiques au plus petit dénominateur commun, par exemple, 31 caractères.
  - Évitez d'utiliser des mots clés réservés dans la norme ANSI et la documentation du fournisseur.
  - Évitez d'utiliser des zones de données spécifiques à des fournisseurs.
  - Évitez les conversions.
  - Vérifiez que la précision et l'échelle sont prises en charge pour tous les fournisseurs.
  - Utilisez des classements cohérents et compatibles.
  - Utilisez une casse cohérente pour les noms, par exemple, tout en minuscules.
- Évaluez le langage de définition de données (DDL) afin de déterminer la portabilité pour la qualification de base de données.
- Évaluez le langage de définition de données afin de déterminer la portabilité pour les types de données en termes de compatibilité, de précision et d'échelle.
- Recherchez dans les instructions SQL natives de vos modèles et rapports toute syntaxe spécifique aux données relationnelles qui peut ne pas être forcément prise en charge.

- Vérifiez l'utilisation des fonctions spécifiques à des fournisseurs.  
Il peut ne pas exister de fonction équivalente ou de fonction commune chez un autre fournisseur. Une fonction commune qui n'est pas prise en charge par la base de données relationnelle peut entraîner un traitement local qui n'était pas effectué précédemment.
- Vérifiez le type de propriété de source de données.  
Si vous modifiez le système de gestion de base de données relationnelle que vous utilisez (par exemple, d'Oracle à SQL Server), modifiez la propriété de type pour la source de données dans Framework Manager.
- Mettez à jour les requêtes de source de données.  
Lorsque vous importez des tables, Framework Manager importe des informations physiques sur les tables et les colonnes qui sont utilisées en interne lors de l'exécution. Par exemple, les informations de classement ne sont mises en correspondance qu'en recréant les tables physiques.
- Testez le modèle déplacé.  
D'autres différences, comme des caractéristiques de performances ou la manière dont les données sont triées en fonction de classements, sont uniquement révélées par des tests.

### **Déplacement de modèles vers des environnements différents à l'aide de fichiers journaux**

Dans IBM Cognos Framework Manager, vous pouvez visualiser et exécuter à nouveau les actions à effectuer sur le projet ou vous pouvez utiliser Script Player pour exécuter à nouveau des transactions en mode de traitement par lots. Un journal des actions est un fichier XML qui contient un ensemble de transactions. Chaque transaction comporte un numéro de séquence et une ou plusieurs actions. Chaque action est constituée d'un nom et de paramètres d'entrée. Certaines actions ont aussi des paramètres de sortie. Le fichier journal des actions est situé dans le dossier du projet.

Par exemple, vous modifiez un projet dans un environnement de test. Au moment de faire passer le projet en production, vous pouvez utiliser des fichiers journaux pour exécuter à nouveau chaque action ou série d'actions que vous avez effectuée dans l'environnement de test pour créer un projet identique dans l'environnement de production.

Il existe deux journaux des actions. Le fichier `log.xml` contient toutes les transactions qui ont été exécutées et enregistrées dans le projet. Ce fichier est créé la première fois que vous enregistrez le projet et existe jusqu'à ce que vous supprimiez le projet. Le fichier temporaire contient les transactions qui ont été exécutées pendant la session en cours, mais pas enregistrées. Il est supprimé lorsque vous fermez le projet.

**Remarque :** Si le script a des dépendances sur le projet existant, vous devez vous assurer que le projet est aligné avec les transactions du script afin de garantir les résultats souhaités.

Un fichier journal volumineux peut avoir un impact sur les performances. Vous pouvez archiver les entrées des fichiers journaux pour réduire leur taille.

#### **Affichage et enregistrement d'un historique des transactions :**

Vous pouvez afficher l'historique des transactions dans un fichier journal des actions et l'enregistrer en tant que script.

## Procédure

1. Cliquez sur **Projet, Afficher l'historique des transactions**.  
**Astuce** : Pour agrandir la boîte de dialogue, cliquez deux fois sur la légende. Cliquez à nouveau deux fois pour restaurer la boîte de dialogue à sa taille d'origine.
2. Cliquez sur les numéros de transaction souhaités.  
**Astuce** : Pour afficher les détails d'une transaction, cliquez sur le signe plus (+) en regard d'un numéro de transaction.
3. Cliquez sur **Save as Script** (Enregistrer en tant que script).
4. Saisissez un nom pour le fichier.
5. Cliquez sur **Enregistrer**. N'enregistrez pas le fichier dans le dossier **logs**.
6. Cliquez sur **Fermer**.

## Réexécution de transactions à partir d'un fichier journal :

Vous pouvez choisir de réexécuter une transaction spécifique ou une combinaison de transactions du fichier journal des actions d'un projet ou d'un segment.



Lorsque vous réexécutez des transactions à partir d'un fichier journal, le lecteur de script applique les commandes du fichier journal au contenu du modèle existant. Des erreurs s'affichent si des objets créés par le fichier journal existent déjà dans le modèle.

Une fois que le script d'un fichier journal a été exécuté avec succès, une copie de sauvegarde du projet d'origine est créée dans le répertoire parent du projet. Si vous souhaitez annuler les transactions exécutées dans le script, vous pouvez utiliser cette copie de sauvegarde pour restaurer le projet à son état d'origine.


Vous devez désactiver ou effacer toutes les commandes entrant en conflit avec le contenu du modèle. Vous pouvez ensuite réexécuter le script. Vous pouvez également utiliser la commande **Synchroniser**, qui permet de commencer avec un modèle vide.

Si vous générez votre propre script en dehors de Framework Manager, les horodatages doivent être en ordre croissant sans doublons.





## Procédure

1. Cliquez sur **Projet, Exécution du script**.
2. Sélectionnez le script voulu et cliquez sur **Ouvrir**.
3. Si vous souhaitez afficher les détails d'une transaction, cliquez sur celle-ci.
4. Définissez le point de départ ou d'arrêt souhaité.
  - Pour définir le point de départ pour l'exécution du script, sélectionnez le script, puis cliquez sur **Définir le point de départ**. Vous pouvez effectuer cette opération à tout moment pour ignorer une instruction ou exécuter des instructions qui ont déjà été exécutées .
  - Pour définir un point d'arrêt pour le script, sélectionnez le script puis cliquez sur **Définir le point d'arrêt** .

Vous pouvez arrêter le script pour le corriger manuellement, puis le redémarrer.

**Astuce :** Pour supprimer le point d'arrêt, cliquez sur **Supprimer le point d'arrêt** 

5. A l'aide des boutons de barre d'outils, sélectionnez l'action à exécuter.

Bouton	Description
	Exécute le script  Lorsqu'une erreur est détectée, vous pouvez cliquer sur ce bouton pour tenter de réexécuter l'instruction qui a échoué.
	Passer à la transaction suivante et exécuter le script jusqu'à la fin
	Exécute la transaction sélectionnée uniquement
	Passer à la transaction suivante et s'arrêter, mais n'exécuter aucune transaction

La fenêtre du projet est mise à jour au fur et à mesure le script est exécuté.

6. Corrigez les erreurs rencontrées par le script en reciblant des objets ou en modifiant le projet temporaire, le cas échéant.  
Pour plus d'informations, voir «Correction des erreurs causées par des objets non valides», à la page 304.
7. Lorsque le script a terminé, cliquez sur **Accepter** pour accepter les modifications ou sur **Annuler** pour annuler les modifications.

**Remarque :** Une fois que vous avez cliqué sur **Accepter** ou **Annuler**, vous ne pouvez plus utiliser **Annuler** et **Rétablir** pour la session en cours.

### Exécution de journaux des actions en mode différé :

*Utilisation de l'interpréteur de script :*

#### Syntaxe

A l'invite de commande, vérifiez que vous accédez à l'emplacement d'installation de BmtScriptPlayer.exe.

Utilisez la syntaxe suivante pour exécuter l'interpréteur de script :

```
BmtScriptPlayer [-c|-m] <projectname> [-a <actionlogname>] [options]
```

où <projectname> est le nom du projet et <actionlogname> correspond au nom du journal des actions.

Par exemple,

```
BmtScriptPlayer -m goSales.cpf -a import.xml
```

#### Options

Vous pouvez spécifier le mode d'exécution de l'interpréteur de script en utilisant les options décrites ci-après.

Si vous travaillez dans un environnement UNIX, vous pouvez être amené à créer un script pour masquer les données d'identification qui sont transmises sur la ligne de commande.

**-a** *FILEPATH*

Appliquer le journal des actions spécifié.

*FILEPATH* correspond au chemin d'accès, nom de fichier compris, vers le fichier journal des actions.

**-b** **NUM**

Exécuter les transactions avec un numéro de séquence égal ou supérieur au nombre indiqué par NUM.

Par défaut, la première transaction est exécutée.

**-c** *FILEPATH*

Créer un nouveau projet.

*FILEPATH* correspond au chemin d'accès, nom de fichier compris, vers le fichier de projet de modèles (.cpf).

Lorsque cette option est utilisée sans indiquer de journal des actions, un modèle vide est créé.

Si le modèle spécifié dans *FILEPATH* existe déjà, il est remplacé de façon silencieuse.

**-e** **NUM**

Exécuter les transactions avec un numéro de séquence égal ou inférieur au nombre indiqué par NUM.

Si l'option n'est pas spécifiée, l'exécution s'arrête à la transaction dont le numéro de séquence est le plus élevé ou dont le numéro de transaction est égal à 9999, selon ce qui se présente en premier. Pour les journaux d'actions qui contiennent des transactions avec des numéros de séquence correspondant à 10 000 et plus, cette option doit être utilisée.

**-g**

Mettre à niveau le modèle (si obligatoire).

Si cette option n'est pas spécifiée alors que le modèle a été créé avec une version précédente, l'exécution se termine.

Lorsque cette option est utilisée sans indiquer de journal des actions, seule la mise à niveau de modèle est effectuée.

**-k** *DIRECTORY*

Indiquer le répertoire d'installation.

**-l** *FILEPATH*

Indiquez le chemin d'accès, nom de fichier compris, vers un fichier qui contient les options à utiliser lorsque l'interpréteur de script est exécuté. En particulier, cette option permet de transmettre des informations de nom d'utilisateur et de mot de passe à l'interpréteur de script sans que celles-ci soient exposées sur la ligne de commande.

**-m** *FILEPATH*

Ouvrir un projet existant.

*FILEPATH* correspond au chemin d'accès, nom de fichier compris, vers le fichier de projet de modèles (.cpf).

**-n**

Ne pas enregistrer le modèle.



Cette option peut être utilisée pour tester des fichiers journaux d'actions.

**-p** *PASSWORD*

Authentifier à l'aide du mot de passe indiqué (si nécessaire).

**-s** *NAMESPACE*

Authentifier à l'aide de l'espace-noms indiqué (si nécessaire).

**-t** *DIRECTORY*

Indiquer le répertoire de modèles.

**-T** *PASSWORD*

Indiquer un passeport de sécurité. Un passeport est une chaîne chiffrée qui permet des conversations sécurisées pour les plug-ins qui en ont besoin.

**-u** *USER*

Authentifier à l'aide du nom d'utilisateur indiqué (si nécessaire).

**-x**

Mettre fin à l'exécution de la série de tests lorsqu'il y a une erreur de transaction.

Par défaut, l'interpréteur de script se termine uniquement lorsque des erreurs graves se sont produites, par exemple un modèle ou un journal des actions non valide, et continue de s'exécuter, même si certaines transactions mineures échouent.

**-y** *PASSPORT*

Authentifier à l'aide du passeport indiqué (si nécessaire).

Cette option remplace toute autre donnée d'identification spécifiée (-s, -p et -u). L'interpréteur de script ignore l'authentification et associe le passeport spécifié à la session.

## Exemples

Le tableau suivant contient quelques exemples de commande de l'interpréteur de script.

Tableau 10. Exemples de commandes de l'interpréteur de script

Commande	Description
<code>BmtScriptPlayer -c &lt;projectname&gt;</code>	Créer un projet.
<code>BmtScriptPlayer -c &lt;projectname&gt; -a &lt;actionlogname&gt;</code>	Créer un projet et appliquer toutes les transactions à partir du journal des actions.
<code>BmtScriptPlayer -c &lt;projectname&gt; -a &lt;actionlogname&gt; -b2 -e20</code>	Créer un projet et appliquer les transactions numérotées de 1 à 20 à partir du journal des actions.
<code>BmtScriptPlayer -m &lt;projectname&gt; -a &lt;actionlogname&gt; -e20</code>	Ouvrir un projet existant et appliquer les transactions numérotées de 1 à 20 à partir du journal des actions.
<code>BmtScriptPlayer -m &lt;projectname&gt; -a &lt;actionlogname&gt; -n</code>	Ouvrir un projet existant et appliquer toutes les transactions à partir du journal des actions. Ne pas enregistrer le projet.

## Correction des erreurs causées par des objets non valides :

Des erreurs peuvent se produire lors de l'exécution des fichiers script ou de la vérification des modèles si un objet qui est référencé par une transaction n'existe plus, ou si vous avez renommé des objets.

Si un objet n'existe plus, ciblez à nouveau l'objet manquant vers un autre objet.

### Utilisation de scripts

Si vous utilisez des scripts et que vous ciblez à nouveau un objet, toutes les transactions de script restantes utilisent le nouvel objet. Si le script s'arrête pour une toute autre raison, vous devez modifier le projet temporaire afin de résoudre le problème.

Remarque : corriger des erreurs en apportant des modifications au projet principal peut produire des résultats imprévisibles. Corrigez toujours les erreurs en modifiant le projet temporaire.

Lorsque des erreurs sont détectées au niveau d'un script, vous pouvez choisir la façon dont vous souhaitez résoudre le problème.

Solution	Action
Ignorer les transactions qui comportent cet objet	Cliquez sur <b>Exclude</b> , et dans la boîte de dialogue <b>Exclude Transactions that Use this Object</b> , sélectionnez le niveau d'exclusion souhaité.  La transaction en cours et toutes les suivantes qui font référence à l'objet exclus sont ignorées. Par exemple, si une transaction tente de créer un pack qui utilise l'objet exclus, le pack n'est pas créé. <b>Remarque :</b> Vous devez corriger les erreurs avant d'ignorer les transactions.
Remplacer cette occurrence et toutes les occurrences suivantes de l'objet	Cliquez sur <b>Replace</b> , et dans la boîte de dialogue <b>Replace Missing Objects</b> , sélectionnez l'option de votre choix.
Corrigez le problème manuellement	Cliquez sur <b>Arrêter</b> , puis corrigez le problème dans le projet temporaire.

### Cibler à nouveau un objet

Si une transaction fait référence à un objet qui n'existe plus, le script s'arrête et une boîte de dialogue s'affiche avec le nom de l'objet qui pose problème. Vous pouvez cibler à nouveau l'objet en cliquant sur **Replace** et en sélectionnant un nouvel objet.

Si un objet manquant s'affiche dans une expression, le script s'arrête et une boîte de dialogue s'affiche avec le nom de l'objet qui pose problème. Vous devez résoudre l'incident manuellement en ouvrant l'expression qui contient l'objet manquant.

## Correction des erreurs détectées au niveau du script

Vous devez corriger les erreurs de script en modifiant le projet temporaire. Corriger des erreurs en apportant des modifications au projet principal peut produire des résultats imprévisibles.

**Astuce:** vous pouvez déplacer ou de réduire la boîte de dialogue **Synchroniser** pour visualiser et modifier le projet.

## Archivage des entrées d'un fichier journal

Au fil du temps, les fichiers journaux d'un projet peuvent devenir volumineux. Un fichier journal volumineux risque d'affecter les performances d'IBM Cognos Framework Manager, en particulier, lors de sa manipulation. Vous pouvez supprimer une partie des entrées d'un fichier journal et les ajouter dans le contenu du fichier archive-log.xml. Framework Manager archive toutes les entrées du fichier journal avant la transaction sélectionnée. Le transaction sélectionné n'est pas archivée.

Les transactions archivées sont disponibles lors de la synchronisation des projets. Pour plus d'informations, voir «Synchronisation des projets».

les transactions archivées ne sont plus visibles dans l'historique des transactions. Ces transactions ne sont pas disponibles lorsque vous branchez ou fusionnez des projets. Par exemple, dans un projet branché, toutes les transactions archivées ne seront pas disponibles lorsque vous refusionnez les branches avec le projet principal.

### Procédure

1. Cliquez sur **Projet, Afficher l'historique des transactions**.
2. Recherchez dans un fichier journal l'entrée correspondant à la transaction qui a eu lieu après la dernière transaction à archiver.  
Toutes les transactions du fichier journal en cours avant la transaction sélectionnée seront archivées.
3. Cliquez sur **Archiver le fichier journal**.

---

## Synchronisation des projets

Vous pouvez utiliser les fichiers journaux d'IBM Cognos Framework Manager pour synchroniser votre projet.

Vous pouvez souhaiter synchroniser votre projet dans les cas suivants :

- Vous avez mis à jour des métadonnées dans un outil de modélisation tiers.
- Vous avez modifié des métadonnées à l'aide d'un outil de modélisation tiers.

Lorsque vous synchronisez votre projet, vous créez un nouveau projet en réexécutant à partir des fichiers journaux toutes les actions que vous avez effectuées dans le projet d'origine.

Plusieurs points sont à prendre en compte avant la synchronisation de projets qui contiennent des modèles segmentés ou des modèles liés.

Si votre source de données est une base de données relationnelle, vous ne pouvez mettre à jour que les sujets de requête et vous n'avez pas besoin d'effectuer une

synchronisation complète de projet. Vous devez effectuer une synchronisation de projet pour synchroniser les modifications effectuées dans une autre source de données.

Si vous importez un sous-ensemble d'une source de données, tous les nouveaux objets qui ont été ajoutés à la source de données ne sont pas inclus lorsque vous effectuez la synchronisation. L'importation d'objets que vous avez spécifiée à l'origine a été enregistrée dans le journal des actions. Lorsque vous effectuez une synchronisation, seuls les objets importés à l'origine sont réimportés.

Vous pouvez utiliser la synchronisation de projet pour exécuter l'historique des actions complet du modèle et mettre à jour les métadonnées du modèle. Vous pouvez également enregistrer des parties du journal des actions dans un fichier script distinct pour une utilisation ultérieure, ou le journal des actions complet dans un fichier script si vous souhaitez générer le même modèle en mode différé. Si des erreurs se produisent lors de l'exécution d'un script de journal des actions, voir «Correction des erreurs causées par des objets non valides», à la page 304.

Après la synchronisation, vous pouvez choisir d'accepter les nouvelles modifications et de créer un nouveau projet, ou de revenir au projet d'origine. Si vous acceptez les nouvelles modifications, le projet d'origine est remplacé.

Comme chacune des actions que vous avez effectuées sur votre projet est réexécutée, la synchronisation peut prendre un certain temps.

Si un objet référencé par une transaction n'existe plus, qu'il ait été renommé ou supprimé, des erreurs sont générées lors de la synchronisation. Par exemple, si vous avez importé une table nommée Produits, puis renommé la table en Nouveaux produits dans votre source de données, vous recevez une erreur lorsque vous synchronisez le projet. La synchronisation ne peut pas détecter que la table appelée Nouveaux produits a été importée précédemment sous un autre nom. Vous devez redéfinir manuellement l'objet pour terminer la synchronisation. Pour plus d'informations sur la correction des erreurs de synchronisation, voir «Correction des erreurs causées par des objets non valides», à la page 304.

**Remarque :** Les journaux des actions d'IBM Cognos ReportNet ne sont pas pris en charge dans cette édition.

Avant de synchroniser un projet, vous devez comprendre l'impact de la synchronisation sur les modèles segmentés et liés. Vous devez également vous assurer que les connexions de sources de données n'ont pas changé et que les sources de données sont en ligne. Vous pouvez vérifier les connexions en testant quelques sujets de requête clés.

## **modèles segmentés**

Un modèle segmenté doit uniquement être synchronisé en synchronisant le projet principal. Les résultats de la synchronisation du projet complet sont écrits dans le fichier journal du projet principal. Vous n'aurez plus la possibilité de synchroniser des segments individuels après la première synchronisation du projet principal.

Si vous travaillez dans le projet principal et que vous modifiez un segment, le fichier journal principal est mis à jour. Si vous travaillez dans le segment et que vous le modifiez, le fichier journal du segment est mis à jour.

Les commandes de synchronisation ne s'exécutent pas nécessairement dans l'ordre affiché dans les fichiers journaux. Ceci s'explique par le fait qu'il est possible de mettre à jour des segments simultanément et que les journaux des actions sont réexécutés en fonction de la date et de l'heure de l'action d'origine. Les commandes peuvent passer d'un fichier journal à l'autre, ce qui rend l'utilisation de fonctions de débogage difficile comme un débogage unique pas à pas.

## Modèles liés

Les fichiers journaux sont situés dans le projet qui est ouvert et non dans le modèle mis à jour.



Si vous ouvrez un projet principal et que vous modifiez un modèle lié, les actions sont journalisées dans le fichier journal du projet principal. Si vous synchronisez ensuite le modèle lié, la modification est perdue car elle ne s'affiche pas dans l'ensemble des fichiers journaux qui ont été utilisés pour la synchronisation.

La synchronisation ne peut être exécutée que sur le projet principal ou un segment autonome. Vous ne pouvez pas synchroniser des projets ou des segments liés dans le projet principal. Si les segments sont mis à jour par le projet lié, la synchronisation peut entraîner des résultats imprévisibles dans le projet principal.


N'utilisez pas la synchronisation de modèle en combinaison avec des projets liés.




## Procédure de synchronisation



### Procédure

1. Cliquez sur **Projet, Synchroniser**.
2. Vous pouvez créer une copie de sauvegarde de votre projet Framework Manager en cochant la case **Sauvegarder le projet dans ce répertoire**.
3. Si vous souhaitez afficher les détails d'une transaction, cliquez sur celle-ci.
4. Définissez le point de départ ou d'arrêt souhaité.
  - Pour définir le point de départ pour l'exécution du script, sélectionnez le script, puis cliquez sur **Définir le point de départ**. Vous pouvez effectuer cette opération à tout moment pour ignorer une instruction ou exécuter des instructions qui ont déjà été exécutées .
  - Pour définir un point d'arrêt pour le script, sélectionnez le script puis cliquez sur **Définir le point d'arrêt** .

Vous pouvez arrêter le script pour le corriger manuellement, puis le redémarrer.

**Astuce :** Pour supprimer le point d'arrêt, cliquez sur **Supprimer le point d'arrêt** .
5. A l'aide des boutons de barre d'outils, sélectionnez l'action à exécuter.

Bouton	Description
	Exécute le script
	Lorsqu'une erreur est détectée, vous pouvez cliquer sur ce bouton pour tenter de réexécuter l'instruction qui a échoué.
	Passes à la transaction suivante et exécute le script jusqu'à la fin

Bouton	Description
	Exécute la transaction sélectionnée uniquement
	Passe à la transaction suivante et s'arrête, mais n'exécute aucune transaction

La fenêtre du projet est mise à jour au fur et à mesure le script est exécuté.

6. Corrigez les erreurs rencontrées par le script en reciblant des objets ou en modifiant le projet temporaire, le cas échéant.  
Pour plus d'informations, voir «Correction des erreurs causées par des objets non valides», à la page 304.
7. Lorsque le script est terminé, cliquez sur **Accepter**.  
Le projet d'origine est remplacé par le contenu du projet temporaire.  
**Astuce** : Pour rétablir le projet à son état précédent, cliquez sur **Annuler**.

---

## Comportement des requêtes

En surveillant et en ajustant le comportement des requêtes dans votre projet IBM Cognos Framework Manager, vous pouvez améliorer les performances de votre modèle.

Vous pouvez effectuer les opérations suivantes :

### Procédure

1. Définissez des restrictions pour réduire les besoins en ressources système et améliorer les performances en vous veillant à ce que les métadonnées d'un pack contiennent les limites appropriées. Par exemple, vous pouvez définir des limites pour la quantité de données extraites ou la durée possible pour une requête. Pour plus d'informations, voir «Restrictions», à la page 309.
2. Indiquez si les cumuls d'agrégation sont calculés en local ou dans la base de données. Dans le cas de métadonnées relationnelles, vous pouvez optimiser les performances en sélectionnant le type approprié pour le traitement de requête. Après l'exécution du rapport initial, en activant la fonction de réutilisation de requête, vous pouvez créer des rapports sans ré-interroger la base de données. Pour plus d'informations, voir «Spécification de l'emplacement du traitement des cumuls d'agrégat», à la page 316.
3. Améliorez les performances en définissant le type de traitement de requête pour déterminer si le traitement SQL est effectué par le serveur de base de données ou en local. Pour plus d'informations, voir «Optimisation des performances en configurant le type de traitement de requête», à la page 317.
4. Définissez le mode transactionnel d'un sujet de requête avec une procédure stockée pour qu'il soit utilisable dans tous les studios. Pour plus d'informations, voir «Définition des modes de transaction», à la page 320.
5. Améliorez les performances grâce à la réutilisation des données mises en cache lors de l'exécution d'un rapport. En réutilisant les données mises en cache, vous pouvez créer un rapport sans ré-interroger la base de données. Pour plus d'informations, voir «Amélioration des performances grâce à la réutilisation des données mises en cache lors de l'exécution d'un rapport», à la page 318.
6. Sélectionnez les ensembles de fonctions spécifiques à des fournisseurs pour les sources de données définies dans le projet. Pour plus d'informations, voir «Sélection d'ensembles de fonctions», à la page 321.

7. Indiquez le comportement des différentes fonctions qui sont basées sur les sources de données du projet. Pour plus d'informations, voir «Qualité de service», à la page 321.
8. Contrôlez et optimisez la façon dont les requêtes sont exécutées en modifiant les propriétés d'une source de données qui a été créée à l'aide de l'assistant de métadonnées dans IBM Cognos Framework Manager. Pour plus d'informations, voir «Contrôle et optimisation des requêtes», à la page 324.

## Restrictions

Utilisez des restrictions pour réduire les besoins en ressources système et améliorer les performances. Vous pouvez définir des restrictions avant de créer des packs pour vous assurer que les métadonnées du pack contiennent les limites spécifiées. Tous les packs publiés ultérieurement utiliseront ces nouveaux paramètres.

Les paramètres de restriction ont la priorité sur les paramètres appliqués au modèle actuellement ouvert (qu'il s'agisse d'un modèle parent ou enfant).

Dans un nouveau projet, les restrictions n'ont pas de valeurs définies dans le modèle. Vous devez ouvrir la fenêtre **Restrictions** et modifier les paramètres si nécessaire. Lorsque vous enregistrez les valeurs dans la fenêtre **Restrictions**, cliquez sur **OK**. Les valeurs pour les restrictions sont alors définies. Vous pouvez également définir des restrictions dans Report Studio. Les paramètres de restriction de Report Studio se substituent aux paramètres de restriction du modèle.

Pour plus d'informations sur les restrictions spécifiques au mode de requête dynamique (DQM), reportez-vous au *Guide sur les requêtes dynamiques*.

### Nombre maximal de tables de rapport

Vous pouvez contrôler le nombre de tables qu'un utilisateur peut extraire dans une requête ou un rapport. Lorsqu'une table est extraite, elle est comptabilisée chaque fois qu'elle s'affiche dans la requête ou le rapport. La limite ne correspond pas au nombre de tables uniques. Si la requête ou le rapport dépasse la limite définie pour le nombre de tables, un message d'erreur s'affiche et la requête ou le rapport est affiché sans données.

La valeur zéro (0) signifie qu'aucune limite n'est définie.

**Remarque :** Cette restriction n'est pas utilisée dans le mode de requête dynamique.

### Nombre maximal de lignes extraites

Vous pouvez définir des limites d'extraction de données en contrôlant le nombre de lignes qui sont renvoyées dans une requête ou un rapport. Les lignes sont comptabilisées au fur et à mesure qu'elles sont extraites.

Si vous exécutez un rapport et que la limite d'extraction de données est dépassée, un message d'erreur s'affiche et la requête ou le rapport est affiché sans données.

Vous pouvez également utiliser cette restriction pour définir des limites pour les données extraites dans un test de sujet de requête de test ou le mode de conception de rapport.

La valeur zéro (0) signifie qu'aucune limite n'est définie.

Si vous externalisez un sujet de requête, ce paramètre est ignoré lorsque vous publiez le modèle. Pour plus d'informations, voir «Dimensions et sujets de requête externalisés», à la page 267.

### Limite temporelle d'exécution de requête

Vous pouvez limiter la durée d'exécution d'une requête. Un message d'erreur s'affiche lorsque le nombre de secondes prédéfini est atteint.

La valeur zéro (0) signifie qu'aucune limite n'est définie.

**Remarque :** Cette restriction n'est pas utilisée dans le mode de requête dynamique.

### Limite des éléments texte de grande taille

Vous pouvez contrôler la longueur des caractères des objets BLOB (Binary Large Object) qu'un utilisateur peut extraire dans une requête ou un rapport. Lorsque la longueur de caractères de l'objet BLOB dépasse la limite définie, un message d'erreur apparaît et la requête ou le rapport est affiché sans données.

La valeur zéro (0) signifie qu'aucune limite n'est définie.

### Jointures externes

Vous pouvez contrôler si des jointures externes peuvent être utilisées dans votre requête ou rapport. Une jointure externe extrait toutes les lignes dans une table, même s'il n'existe pas de ligne correspondante dans une autre table. Ce type de jointure peuvent produire des requêtes et des rapports de très grande taille consommant beaucoup de ressources.

Les restrictions sont définies pour refuser les jointures externes par défaut. Par exemple, des jointures externes ne sont pas générées automatiquement lorsque vous testez un élément de requête dans Framework Manager.

Le code SQL est généré automatiquement lorsque vous

- exécutez un rapport,
- testez un élément de requête ou une relation dans Framework Manager,
- créez un sujet de requête de modèle basé sur d'autres objets.

Si vous conservez la valeur **Refuser**, vous êtes averti uniquement si vous créez dans l'onglet Diagramme une relation qui inclut les jointures externes. Vous n'êtes pas informé si vous créez une relation dans un sujet de requête de source de données qui contient des jointures externes.

Si vous définissez la restriction sur **Autoriser**, les relations dimension-fait sont modifiées de jointures internes en jointures externes.

La restriction de jointure externe ne s'applique pas dans les cas suivants :

- Le code SQL est généré par d'autres moyens. Si vous définissez cette restriction sur **Refuser**, cela ne s'applique pas au code SQL permanent trouvé dans un sujet de requête de source de données, que le code SQL ait été généré lors de l'importation, ait été saisi manuellement ou soit basé sur des objets existants.



- Framework Manager doit générer une jointure externe pour créer une requête liée. Une requête liée est une requête qui combine localement les résultats de plusieurs sous-requêtes en utilisant une jointure externe traitée en local.

**Remarque :** Cette restriction n'est pas applicable aux sources de données SAP BW.

**Remarque :** Cette restriction n'est pas utilisée dans le mode de requête dynamique.

## Jointures inter-produits

Vous pouvez contrôler si des jointures inter-produits peuvent être utilisées dans votre requête ou rapport. Une jointure inter-produits extrait des données de tables sans jointures. Ce type de jointure peut prendre beaucoup de temps pour extraire les données.

La valeur par défaut de cette restriction est **Refuser**. Sélectionnez **Autoriser** pour autoriser les jointures inter-produits.

## Traitement des raccourcis

Vous pouvez contrôler la façon dont les raccourcis sont traités par le logiciel IBM Cognos.

Lorsque vous ouvrez un modèle d'une édition précédente, la restriction **Traitement des raccourcis** a pour valeur **Automatique**. **Automatique** est un raccourci qui se trouve dans le même répertoire que sa cible se comporte comme un alias ou une instance indépendante. En revanche, un raccourci qui se trouve ailleurs dans le modèle se comporte comme une référence au raccourci d'origine. Lorsque vous créez un nouveau modèle, la valeur de la restriction **Traitement des raccourcis** est toujours **Jointure explicite**.

Lorsque vous définissez la restriction sur **Jointure explicite**, le comportement du raccourci est dicté par la propriété **Traiter en tant que**. Si la restriction **Traitement des raccourcis** est définie sur **Automatique**, vérifiez le modèle et affectez la valeur **Jointure explicite** à la restriction au moment de la réparation. Ceci modifie tous les raccourcis en leur affectant la valeur correcte à partir de la propriété **Traiter en tant que** selon les règles suivies par le paramètre **Automatique**.

La restriction **Traitement des raccourcis** a la priorité sur la propriété **Traiter en tant que**. Par exemple, si la restriction est définie sur **Automatique**, le comportement du raccourci est déterminé par l'emplacement du raccourci par rapport à sa cible, quelle que soit la valeur de la propriété **Traiter en tant que**.

## Syntaxe de jointure SQL

Vous pouvez contrôler la façon dont le code SQL est généré pour les jointures internes dans un modèle en sélectionnant l'un des paramètres suivants :

- Si la restriction est définie sur **Déterminé par le serveur**, le fichier CQEConfig.xml est utilisé pour déterminer la valeur de la restriction. S'il n'existe aucun fichier CQEConfig.xml actif ni aucune entrée de paramètre pour la restriction dans le fichier CQEConfig.xml, le paramètre **Jointure implicite** est utilisé.
- Le paramètre **Jointure implicite** utilise la clause where.  
Par exemple :

```
SELECT publishers.name, publishers.id,
books.title FROM publishers, books WHERE publishers.id
= books.publisher_id ORDER BY publishers.name, books.title;
```

- Le paramètre **Jointure explicite** utilise la clause from avec les mots clés inner join dans un prédicat on.

Par exemple :

```
SELECT
publishers.name, publishers.id,
books.title FROM publishers INNER JOIN books ON publishers.id
= books.publisher_id ORDER BY publishers.name, books.title;
```

Vous pouvez définir le type de jointure sur la propriété de la requête dans Report Studio pour redéfinir la valeur de cette restriction.

Quel que soit le paramètre que vous utilisez pour cette restriction, le paramètre **Jointure explicite** est utilisé pour les jointures externes gauches, les jointures externes droite et les jointures externes complètes.

Cette restriction n'a aucun impact sur le code SQL saisi.

## Regroupement des attributs de mesure (éléments de requête)

Si la restriction est définie sur **Déterminé par le serveur**, le fichier CQEConfig.xml est utilisé pour déterminer la valeur de la restriction. S'il n'existe aucun fichier CQEConfig.xml actif ni aucune entrée de paramètre pour la restriction dans le fichier CQEConfig.xml, le paramètre **Désactivé** est utilisé.

Le paramètre **Désactivé** empêche l'agrégation de la mesure pour les attributs. Il s'agit du comportement par défaut. Par exemple :

```
select Product.Product_line_code as Product_line_code,
Order_method.Order_method_code as Order_method_code,
XSUM(Sales.Quantity for Product.Product_line_code) as Quantity from ...
```

Le paramètre **Activé** autorise l'agrégation de la mesure pour les attributs.

**Remarque :** Il s'agit du comportement par défaut pour les versions d'IBM Cognos Framework Manager antérieures à la version 8.3.

```
select Product.Product_line_code as Product_line_code,
Order_method.Order_method_code as Order_method_code,
XSUM(Sales.Quantity for Order_method.Order_method_code,
Product.Product_line_code)
as Quantity
from ...
```

Dans ces exemples, Order\_method.Order\_method\_code as Order\_method\_code définit l'attribut de mesure.

La syntaxe XSUM définit la mesure agrégée.

## Génération SQL pour les attributs de niveau

Vous pouvez contrôler l'utilisation de l'agrégat minimum dans le code SQL généré pour les attributs d'un niveau (légende du membre).

Si la restriction est définie sur **Déterminé par le serveur**, le fichier CQEConfig.xml est utilisé pour déterminer la valeur de la restriction. S'il n'existe aucun fichier CQEConfig.xml actif ni aucune entrée de paramètre pour la restriction dans le fichier CQEConfig.xml, le paramètre **Minimum** est utilisé.

Le paramètre **Minimum** génère l'agrégat minimum pour l'attribut. Ce paramètre garantit l'intégrité des données s'il existe un risque d'enregistrements en double. Par exemple :

```
select XMIN(Product.Product_line
for Product.Product_line_code) as Product_line, //level attribute
Product.Product_line_code as Product_line_code
from
(...) Product
```

Le paramètre **Regrouper par** ajoute les attributs du niveau à la clause group by sans agrégation pour l'attribut. La clause distinct applique une instruction group by sur tous les éléments de la liste de projection. Le paramètre **Regrouper par** est utilisé si les données n'ont pas d'enregistrements en double. Il peut améliorer l'utilisation de vues matérialisées et se traduire par une amélioration des performances. Par exemple :

```
select distinct
Product.Product_line as Product_line, //level attribute
,Product.Product_line_code
as Product_line_code
from(...) Product
```

**Remarque :** Cette restriction n'est pas utilisée dans le mode de requête dynamique.

## Génération SQL pour les attributs de déterminant

Vous pouvez contrôler l'utilisation de l'agrégat minimum dans le code SQL généré pour les attributs d'un déterminant avec la propriété Regrouper par activée.

Si la restriction est définie sur **Déterminé par le serveur**, le fichier CQEConfig.xml est utilisé pour déterminer la valeur de la restriction. S'il n'existe aucun fichier CQEConfig.xml actif ni aucune entrée de paramètre pour la restriction dans le fichier CQEConfig.xml, le paramètre **Minimum** est utilisé.

Le paramètre **Minimum** génère l'agrégat minimum pour l'attribut. Ce paramètre garantit l'intégrité des données s'il existe un risque d'enregistrements en double. Par exemple :

```
select PRODUCT_LINE.PRODUCT_LINE_CODE
as Product_line_code,
XMIN(PRODUCT_LINE.PRODUCT_LINE_EN
for PRODUCT_LINE.PRODUCT_LINE_CODE)
as Product_line //attribute
from
great_outdoors_sales..GOSALES.PRODUCT_LINE PRODUCT_LINE
group by
PRODUCT_LINE.PRODUCT_LINE_CODE //key
```

Le paramètre **Regrouper par** ajoute les attributs des déterminants à la clause group by sans agrégation pour l'attribut. Ce paramètre est utilisé si les données n'ont pas d'enregistrements en double. Il peut améliorer l'utilisation de vues matérialisées et se traduire par une amélioration des performances. Par exemple :

```
select
PRODUCT_LINE.PRODUCT_LINE_CODE as Product_line_code,
PRODUCT_LINE.PRODUCT_LINE_EN as Product_line //attribute
from
great_outdoors_sales..GOSALES.PRODUCT_LINE PRODUCT_LINE
group by
PRODUCT_LINE.PRODUCT_LINE_CODE //key
PRODUCT_LINE.PRODUCT_LINE_EN //attribute
```

## Syntaxe de paramètre SQL

Cette restriction indique si le code SQL généré utilise des repères de paramètre ou des valeurs littérales.

Si la restriction est définie sur **Déterminé par le serveur**, le fichier CQEConfig.xml est utilisé pour déterminer la valeur de la restriction. S'il n'existe aucun fichier CQEConfig.xml actif ni aucune entrée de paramètre pour la restriction dans le fichier CQEConfig.xml, le paramètre **Repère** est utilisé.

Vous pouvez redéfinir la valeur de cette restriction dans Report Studio.

Les applications SQL dynamiques ont la possibilité de préparer des instructions qui incluent des repères dans le texte indiquant que la valeur sera fournie ultérieurement. Il s'agit de la méthode la plus efficace lorsque la même requête est utilisée de nombreuses fois avec des valeurs différentes. Cette technique réduit le nombre d'analyses syntaxiques complètes devant être effectuées par une base de données sur une instruction SQL et augmente la réutilisation des instructions mises en cache. Cependant, lorsque des requêtes explorent de grandes quantités de données avec des instructions plus complexes, elles ont moins de chances de correspondre à d'autres requêtes. Dans ce cas, l'utilisation de valeurs littérales au lieu de repères peut se traduire par une amélioration des performances.

## Autoriser la portabilité du modèle amélioré lors de l'exécution

Cette restriction est sélectionnée lors de la mise à niveau initiale d'un modèle Cognos ReportNet 1.x. Elle empêche une application rigide des types de données de sorte qu'un modèle IBM Cognos puisse fonctionner comme un modèle ReportNet 1.x jusqu'à ce que vous mettiez à jour les types de données dans les métadonnées. Après avoir vérifié que le modèle a été mis à niveau avec succès, désélectionnez cette restriction.

Outre la mise à niveau initiale, il existe des utilisations limitées pour cette restriction. Par exemple, vous avez créé un modèle avec une source de données et vous souhaitez l'exécuter sur une autre source de données. La nouvelle source de données doit avoir une structure similaire à celle de la source de données d'origine et le schéma de base de données doit être le même pour les deux sources de données. Si vous sélectionnez cette restriction, IBM Cognos BI extrait les métadonnées de la source de données et les met en cache au lieu d'utiliser les métadonnées déjà mises en cache dans le modèle. Désélectionnez cette restriction lorsque vous avez fini de modifier et tester le modèle avec la nouvelle source de données.

Si vous n'utilisez pas cette restriction, vous devez vous assurer que les métadonnées suivantes sont les mêmes dans la source de données d'origine et la nouvelle source de données :

- Nom de séquence de classement
- Niveau de classement
- Jeu de caractères
- Acceptabilité des valeurs indéfinies (NULL)
- Précision
- Echelle
- Longueur de colonne
- Type de données

## Autoriser l'utilisation du cache local

Sélectionnez cette restriction pour indiquer que tous les rapports basés sur ce modèle utiliseront les données mises en cache. Pour un nouveau modèle, cette restriction est activée par défaut.

Ce paramètre affecte tous les rapports qui utilisent le modèle. Utilisez Report Studio si vous souhaitez qu'un rapport utilise un paramètre différent que celui du modèle. Pour plus d'informations, voir «Amélioration des performances grâce à la réutilisation des données mises en cache lors de l'exécution d'un rapport», à la page 318.

## Autoriser la génération dynamique des informations de dimension

Cette restriction est sélectionnée lors de la mise à niveau initiale d'un modèle ReportNet 1.x. Elle permet un comportement cohérent avec ReportNet 1.x en dérivant des sortes d'informations de dimension à partir des relations, des informations de clé et des informations d'index de la source de données.

## Utiliser la clause WITH lors de la génération de SQL

Vous pouvez choisir d'utiliser la clause `With` avec IBM Cognos SQL si votre source de données prend en charge cette clause.

La clause `With` est activée pour les modèles créés dans IBM Cognos BI. Pour les modèles mis à niveau, elle est désactivée à moins qu'elle ait été explicitement activée dans le modèle Cognos ReportNet avant la mise à niveau.

## Supprimer les valeurs nulles pour les sources de données SAP BW

Vous pouvez contrôler si les valeurs nulles sont supprimées ou non par un rapport ou une analyse qui utilise le pack publié. La restriction est également appliquée aux résultats de test pendant la session Framework Manager en cours. Elle n'est prise en charge que par les sources de données SAP BW.

Certaines requêtes peuvent être volumineuses car les valeurs nulles ne sont pas supprimées par filtrage. La suppression des valeurs nulles permet de supprimer une ligne ou une colonne dont toutes les valeurs sont nulles (vides). Cette opération est effectuée par SAP BW. Elle réduit la quantité de données transférées vers les produits client IBM Cognos et améliore les performances.

Par défaut, les valeurs nulles sont supprimées. Si vous désélectionnez cette restriction, les valeurs nulles ne sont pas supprimées.

Il existe une propriété appelée **Suppression** dans Report Studio qui se substitue à cette restriction. Si la propriété **Suppression** est définie sur **Aucun**, les valeurs nulles sont incluses dans l'ensemble de résultats même si la restriction est définie sur Supprimer les valeurs nulles.

Remarque : Cette restriction n'est pas appliquée lors de la création de fichiers CSV. Par conséquent, les fichiers CSV incluent des valeurs nulles si elles existent dans les données.

## Traitement du modèle entier lors d'une publication

Un pack publié comprend des objets de modèle sélectionnés lors de sa création. En outre, ces objets de modèle sont analysés pour identifier et inclure les objets dépendants dans le pack.

Dans un modèle complexe ou de très grande taille, l'analyse peut durer très longtemps. Pour raccourcir le délai de publication, définissez cette restriction pour ignorer cette étape d'analyse et écrire le modèle complet dans le magasin de contenu. Le pack résultant peut être plus volumineux car le modèle complet est publié et non seulement les objets requis. Cependant, le temps nécessaire à la publication devrait être réduit.

## Nombre maximal de sources de données externes pouvant être fusionnées avec un modèle

Pour utiliser des données externes, les utilisateurs de rapport importent leurs données dans un pack existant. Cette restriction contrôle le nombre de fichiers de données externes pouvant être importés.

La valeur par défaut est 1.

Pour plus d'informations sur les sources de données externes, reportez-vous au *Guide d'utilisation* d'IBM Cognos Report Studio.

## Taille maximale du fichier de données externes (Ko) :

Pour utiliser des données externes, les utilisateurs de rapport importent leurs données dans un pack existant. Cette restriction contrôle la taille de chaque fichier de données externes.

Par défaut, la taille de fichier maximale que les utilisateurs du rapport peuvent importer est de 2560 ko.

Pour plus d'informations sur les sources de données externes, reportez-vous au *Guide d'utilisation* d'IBM Cognos Report Studio.

## Nombre maximal de lignes de données externes

Pour utiliser des données externes, les utilisateurs de rapport importent leurs données dans un pack existant. Cette restriction contrôle le nombre de lignes pouvant exister dans chaque fichier de données externes.

Par défaut, le nombre maximal de lignes que les utilisateurs peuvent importer est 20000.

Pour plus d'informations sur les sources de données externes, reportez-vous au *Guide d'utilisation* d'IBM Cognos Report Studio.

## Spécification de l'emplacement du traitement des cumuls d'agrégat

La propriété Traitement du cumul pour des sources de données détermine le mode de calcul des cumuls d'agrégat au-dessus du niveau de détail dans le rapport. La valeur par défaut est Local si le traitement de requête local est activé et est sinon défini sur Base de données.

**Remarque :** Cette propriété n'est pas applicable aux sources de données SAP BW.

Les options possibles pour cette propriété sont les suivantes :

- Non spécifié  
Le cumul d'agrégation n'est pas spécifié.
- Local  
Tous les cumuls d'agrégation sont calculés localement (dans le serveur de rapports) à l'aide d'un agrégat cumulatif (par exemple, RSUM). Les agrégats cumulatifs échelonnent le coût de ce calcul au fur et à mesure que les données sont extraites. Utilisez cette option si l'ordinateur local comporte plus de ressources inactives que l'ordinateur de base de données, ou si vous avez constaté qu'il s'agit de la méthode la plus rapide.
- Base de données  
Les cumuls d'agrégation sont calculés par le logiciel de base de données sous-jacent si c'est possible. Sinon, ils sont calculés en local (à condition que le traitement de requête local soit activé). Des agrégats cumulatifs sont utilisés mais le coût est engagé par le serveur de base de données et non au serveur de rapports. Utilisez cette option si l'ordinateur de base de données comporte plus de ressources inactives que l'ordinateur local, ou si vous avez constaté qu'il s'agit de la méthode la plus rapide.
- Étendu  
Tous les cumuls d'agrégation sont calculés par le serveur de base de données à l'aide d'un agrégat étendu (par exemple, XSUM). Les agrégats étendus engagent la totalité du coût de ce calcul à l'avance. En règle générale, il s'agit de la méthode la plus rapide, mais uniquement lorsque la base de données est configurée pour tirer parti des vues matérialisées. Pour les bases de données pour lesquelles la fonctionnalité OLAP est prise en charge, cette option est convertie en les fonctions d'agrégation OLAP appropriées.

### Procédure

1. Dans l'**afficheur de projet**, cliquez sur la source de données à modifier.
2. Dans la sous-fenêtre **Propriétés**, dans la zone de liste **Traitement du cumul**, sélectionnez le type de traitement de cumul souhaité.

## Optimisation des performances en configurant le type de traitement de requête

La propriété de traitement de requête pour des sources de données détermine si le traitement SQL est effectué par le serveur de base de données ou en local. Dans le cas de métadonnées relationnelles, vous pouvez optimiser les performances en sélectionnant le type approprié pour le traitement de requête.

Il existe deux types de traitement de requête :

- Traitement local limité  
Le serveur de base de données effectue le plus de traitement et d'exécution SQL possible. Toutefois, certains rapports ou sections de rapport utilisent le traitement SQL local.
- Base de données seulement  
Le serveur de base de données effectue la totalité du traitement et de l'exécution SQL. Une erreur s'affiche si des rapports ou des sections de rapport nécessitent un traitement SQL local.

Bien que le serveur de base de données permette généralement une exécution plus rapide du code SQL et des rapports, le traitement local peut s'avérer parfois nécessaire. Par exemple, choisissez un traitement local limité si vous souhaitez créer des jointures de bases de données ou si vous souhaitez permettre à vos utilisateurs d'avoir recours à des fonctions SQL99 non prises en charge.

Certaines requêtes complexes nécessitent un traitement local limité, c'est le cas par exemple d'une requête qui doit générer une clause At pour éviter un double comptage.

### **Traitement de requête pour le mode de requête dynamique**

En mode de requête dynamique, les paramètres de traitement de requête ont une incidence uniquement sur les sources de données SAP BW.

- **Traitement local limité**  
Le traitement local limité n'est pas pris en charge pour l'instant.
- **Base de données seulement**  
Un traitement minimal, voir aucun traitement de requête, n'est exécuté par le serveur de rapports. Le traitement local n'a lieu que si la base de données ne peut pas traiter la requête. Envisagez cette option uniquement si les performances de votre rapport sont inacceptables avec le paramètre par défaut et deviennent utilisables avec ce paramètre. Gardez à l'esprit que les résultats peuvent être modifiés avec ce paramètre. Vérifiez avec soin que les résultats restent corrects.

Dans l'**afficheur de projet**, cliquez sur la source de données à modifier. Dans la sous-fenêtre **Propriétés**, dans la zone de liste **Traitement de requête**, cliquez sur **Traitement local limité** ou **Base de données seulement**

## **Amélioration des performances grâce à la réutilisation des données mises en cache lors de l'exécution d'un rapport**

Lorsque vous exécutez un rapport, la demande de requête est envoyée à la base de données et l'ensemble de résultats est renvoyé. Après l'exécution initiale du rapport, vous pouvez décider de modifier le rapport. Souvent, le rapport peut être créé sans interroger à nouveau la base de données. Pour bénéficier de cet avantage, activez la fonction de réutilisation des requêtes.

Lorsque la réutilisation des requêtes est activée et que vous exécutez un rapport pour la première fois, la requête est stockée dans le cache sur le serveur de rapport. En outre, certaines ressources de source de données peuvent ne pas être disponibles tant que la transaction qui les utilise n'a pas été libérée. Le délai d'attente par défaut en cours est de 60 minutes. Si certaines activités de base de données impliquent la modification d'objets de base de données, vous devez attendre la fin du délai d'attente pour terminer l'exécution, ou vous pouvez désactiver la réutilisation des requêtes pour les rapports.

La première fois que le rapport est exécuté et que le cache est créé, cela peut entraîner un léger impact négatif sur le temps de réponse. Par contre, l'utilisateur du rapport bénéficie d'une amélioration des performances à chaque exécution ultérieure du rapport avec un temps de réponse pouvant être raccourci à hauteur de 80 %. Cette amélioration des performances a lieu car le rapport n'a pas besoin de ré-interroger la base de données. En outre, la diminution des requêtes vers la base de données se traduit par une amélioration des performances globales du système, ce qui a un impact positif pour tous les utilisateurs.



La réutilisation des requêtes peut être définie sur le modèle ou sur des rapports individuels. Pour indiquer que tous les rapports utilisant un modèle donné doivent faire appel aux données en cache, activez la restriction **Autoriser l'utilisation du cache local** sur le modèle dans IBM Cognos Framework Manager et republiez le modèle. Par défaut, ce paramètre affecte tous les rapports qui utilisent ce modèle, y compris les analyses qui sont exécutées en tant que rapports dans IBM Cognos Viewer.

## Réutilisation des requêtes dans IBM Cognos Viewer

Si vous souhaitez qu'un rapport utilise un paramètre différent que celui du modèle, vous pouvez le définir dans IBM Cognos Report Studio. Dans la sous-fenêtre **Propriétés**, modifiez la propriété **Utiliser le cache local**. Définissez la propriété sur **Non** si vous souhaitez toujours exécuter la requête, ou sur **Oui** pour utiliser les résultats mis en cache. Si vous voulez que le rapport utilise les mêmes valeurs que le modèle, modifiez le paramètre sur **Par défaut**.

La modification de la propriété **Utiliser le cache local** pour un rapport n'affecte pas les autres rapports.

## Réutilisation des données mises en cache dans Query Studio

IBM Cognos Query Studio réutilise les données mises en cache sous certaines conditions. Si la réutilisation des requêtes est activée sur le modèle et qu'un sous-ensemble de l'ensemble de données mis en cache répond aux besoins de l'action, le rapport utilise les données du cache. Par exemple, des modifications apportées au rapport, telles que l'ajout d'un filtre ou la suppression d'une colonne, peuvent modifier les données du rapport, mais un sous-ensemble des données mises en cache peut quand-même satisfaire la demande.

Si la réutilisation des requêtes est désactivée et que les données mises en cache répondent aux besoins de l'action sans modification, le rapport utilise quand-même les données mises en cache. Par exemple, lorsque le format du rapport est modifié, l'ensemble de données précédent est utilisé même si la réutilisation des requêtes est désactivée. Cette fonction est appelée réutilisation du curseur. La réutilisation du curseur est utilisée lorsque les données mises en cache peuvent satisfaire la demande sans modification.

Les rapports créés dans Query Studio utilisent toujours le même paramètre que le modèle. Si la réutilisation des requêtes est activée pour le modèle, le rapport tente d'utiliser les données mises en cache.

## Utilisation ou non de la réutilisation des requêtes dans votre environnement

Pour savoir s'il faut activer ou non la réutilisation des requêtes, tenez compte des points suivants :

- Si la plupart des utilisateurs de rapport exécutent des rapports interactivement mais ne les exécutent qu'une seule fois, la mise en cache des données ne vous permettra pas d'améliorer beaucoup les performances.

**Remarque :** Quels que soient les paramètres de réutilisation des requêtes, les rapports qui s'exécutent en mode de traitement par lots ne mettent pas de données en cache.

- La taille du cache peut avoir un impact sur l'évolutivité. Par exemple, si un rapport comporte un ensemble de résultats volumineux, la taille du cache sera également importante. Ceci doit être pris en compte lors de la définition de la taille et de la configuration de votre environnement de serveur.

## Définition des modes de transaction

Vous pouvez définir le type de transaction utilisé par les sujets de requête avec des procédures stockées.

Par défaut, un sujet de requête qui contient une procédure stockée est exécuté dans une transaction en lecture seule. Cependant, la procédure stockée peut inclure des opérations qui nécessitent une transaction de type lecture/écriture. Les propriétés **Mode d'accès de la transaction** et **Mode de l'instruction de transaction** permettent l'utilisation dans tous les autres studios des sujets de requête exécutés dans des transactions de type lecture/écriture.

La propriété **Mode d'accès de la transaction** des sources de données définit le mode d'accès des nouvelles transactions. Les différents mode d'accès sont les suivants :

- Non spécifié - Les nouvelles transactions sont définies dans le mode par défaut.
- Lecture seule - Les nouvelles transactions sont définies dans le mode lecture seule.
- Lecture-écriture - Les nouvelles transactions sont définies dans le mode lecture/écriture.

Modes par défaut : En mode de requête dynamique, le mode par défaut est Non spécifié. En mode de requête compatible, le mode par défaut est Lecture seule.

Limitation : En mode de requête dynamique, la propriété **Mode d'accès de la transaction** n'est prise en charge que pour les procédures stockées basées sur des sujets de requête.

La propriété **Mode de l'instruction de transaction** définit l'action qui suit la fin d'une transaction. Les actions suivantes sont possibles :

- Non spécifiée - Définit l'action par défaut qui suit la fin d'une transaction.
- Annulation - La transaction est annulée quand elle prend fin.
- Validation - La transaction est validée quand elle prend fin.
- Validation automatique - La transaction est validée automatiquement quand elle prend fin.

Actions par défaut : Annulation. Si la base de données sou-jacente prend en charge la validation automatique, l'action par défaut est Validation.

Limitation : La propriété **Mode de l'instruction de transaction** est ignorée par le mode de requête dynamique.

Dans l'**afficheur de projet**, cliquez sur la source de données à modifier. Cliquez sur les paramètres choisis dans les zones de liste **Mode d'accès de la transaction** et **Transaction Mode de l'instruction de transaction** de la sous-fenêtre **Propriétés**.

## Sélection d'ensembles de fonctions

Une collection de fonctions spécifiques à des fournisseurs est appelée ensemble de fonctions. Lorsque vous créez un projet qui contient des métadonnées relationnelles, l'éditeur d'expression répertorie les jeux de fonctions pour tous les fournisseurs disponibles. Cependant, vous pouvez restreindre les ensembles de fonctions de façon à ne répertorier que les fournisseurs que vous souhaitez utiliser dans votre projet. Vous personnalisez l'ensemble de fonctions en identifiant le fournisseur spécifique pour chaque source de données définie dans le projet.

Vous pouvez utiliser des fonctions que vous avez définies dans votre source de données relationnelles dans IBM Cognos Framework Manager. Si vous avez importé les fonctions définies par l'utilisateur, elles sont répertoriées dans Framework Manager pour faciliter la sélection. Si vous ne les avez pas importées, vous pouvez entrer le nom de la fonction dans une expression. Si la fonction doit être qualifiée, vous devez l'importer dans Framework Manager.

Parfois, les fonctions spécifiques à des fournisseurs ont été créées sur des schémas avec les droits d'accès importants. Vous ne pouvez pas utiliser ces fonctions dans IBM Cognos Software sur un schéma avec des droits d'accès limités si les deux schémas se trouvent sur la même instance de base de données.

Si une fonction non reconnue est entrée dans un rapport, la fonction est considérée comme étant native. Pour plus d'informations, voir «Code SQL natif», à la page 103.

**Remarque :** Lorsque vous créez un projet qui contient des métadonnées SAP BW, Framework Manager ne répertorie automatiquement que les fonctions qui s'appliquent aux données SAP BW.

### Procédure

1. Cliquez sur **Projet, Liste de fonctions de projet**.
2. Cochez la case **Définir la liste de fonctions selon le type de source de données**.  
**Astuce :** Pour désactiver ce filtre, cochez la case **Inclure tous les ensembles de fonctions**.
3. Dans la page **Ensemble de fonctions**, cliquez sur la ligne de source de données appropriée.
4. Dans la liste déroulante de la zone **Ensemble de fonctions**, sélectionnez l'ensemble de fonctions à utiliser avec cette source de données.
5. Répétez les étapes 2 à 4 jusqu'à ce que vous ayez terminé.
6. Cliquez sur **OK**.

## Qualité de service

Avec IBM Cognos Framework Manager, vous pouvez interroger n'importe quelle combinaison de types de source de données, mais toutes les sources de données ne prennent pas en charge les fonctions de la même façon. L'indicateur de qualité de service fournit à vos utilisateurs et vous un indice visuel sur le comportement des différentes fonctions lorsqu'elles sont utilisées avec les sources de données dans le modèle.

Chaque fonction spécifiée dans votre source de données peut avoir une qualité de service différente selon le type de source de données utilisé. Pour chaque fonction de requête qui n'est pas associée à la même qualité de service dans les différents

packs, vous pouvez remplacer le niveau de service et ajouter un texte pour décrire la situation particulière dans ce modèle. Les utilisateurs peuvent afficher les indicateurs de qualité de service et la description spécifique du contexte, et utiliser ces informations pour déterminer quelles fonctions utiliser dans les rapports.

## **Impact du remplacement de l'indicateur de qualité de service**

IBM Cognos Framework Manager détermine la qualité de service pour les fonctions selon le type de source de données. En tenant compte du contexte du modèle, vous pouvez remplacer la qualité de service déterminée par le produit. En remplaçant la qualité de service, vous fournissez des instructions aux utilisateurs. Cela ne modifie pas le niveau de prise en charge de cette fonction dans votre source de données.

Lorsqu'un pack est créé en combinant des sous-packs, les substitutions de qualité de service du pack parent ont la priorité. S'il n'existe aucune substitution parent, la qualité de service des packs enfant est agrégée.

## **Prenez en compte les utilisateurs**

L'objectif premier est de fournir aux utilisateurs des informations suffisantes pour leurs besoins métier sans pour autant semer la confusion. Si les utilisateurs ne peuvent pas prendre de décisions sur les fonctions à utiliser en se basant sur les indicateurs de qualité de service, il vous faudra peut-être publier des packs distincts pour les différents groupes d'utilisateurs. Si les utilisateurs ont besoin d'accéder à des fonctions dont la qualité de service est inférieure à Non contraint, vous devez indiquer les restrictions de ces fonctions lorsque vous définissez la qualité de service.

## **Impacts sur les performances**

Les indicateurs de qualité de service n'ont aucun impact direct sur les performances des requêtes. Ils sont destinés à vous conférer un certain contrôle sur les fonctions qui sont disponibles pour utilisation. Vous pouvez alors empêcher les utilisateurs d'utiliser des fonctions qui pourraient se traduire par des requêtes trop longues et des requêtes qui échouent.

Il convient de noter que si vous faites appel à des fonctions qui ne sont pas disponibles dans votre source de données, Framework Manager tente de compenser en utilisant un traitement local sur le serveur de rapport. Cela peut avoir une incidence sur les performances des requêtes car le travail est accompli sur le serveur de rapport et non sur le serveur de sources de données.

Dans certains cas, le traitement local peut nécessiter l'extraction de davantage de données du serveur de sources de données, ce qui a un impact sur le serveur de sources de données et le réseau. Par exemple, les fonctions OLAP ne sont pas disponibles dans une source de données relationnelles. Si vous tentez d'utiliser des fonctions OLAP avec une source de données relationnelle, Framework Manager utilise les informations de dimension de la source de données afin de générer un cube local et d'exécuter les fonctions OLAP sur ce cube. Cela nécessite l'extraction des informations de dimension depuis le serveur de sources de données et un traitement supplémentaire sur le serveur de rapport.

## Définition des indicateurs de qualité de service

La qualité de service pour une fonction est spécifiée au niveau source de données et peut être définie pour une fonction individuelle ou pour toutes les fonctions d'un pack.

Pour en savoir davantage sur les fonctions individuelles, reportez-vous à la section Annexe C, «Utilisation de l'éditeur d'expression», à la page 403.

Les indicateurs de qualité de service sont les suivants :

- Non disponible (X)  
Cette fonction n'est disponible pour aucune des sources de données du pack.
- Disponibilité limitée (!!)  
La fonction n'est pas disponible pour certaines sources de données du pack.
- Support limité (!)  
La fonction est disponible pour toutes les sources de données du pack, mais n'est pas naturellement prise en charge pour cette source particulière. IBM Cognos Software utilise une approximation locale de cette fonction. Toutefois, cela peut nuire aux performances et il est possible que vous n'obteniez pas le résultat escompté.
- Non contraint (coche)  
La fonction est disponible pour toutes les sources de données.

Si le modèle contient plusieurs sources de données, les valeurs de qualité de service sont agrégées selon les règles suivantes :

- Si la qualité de service est définie sur Non contraint, Support limité ou Disponibilité limitée dans une source de données, et sur Non disponible dans une autre source de données, la qualité de service pour cette fonction devient Disponibilité limitée.
- Dans tous les autres cas, le plus petit dénominateur commun est utilisé. Par exemple, si la qualité de service est définie sur Non contraint dans une source de données, et sur Support limité dans une autre source de données, la qualité de service pour cette fonction devient Support limité. Si la qualité de service est définie sur Support limité dans une source de données, et sur Disponibilité limitée dans une autre source de données, la qualité de service pour cette fonction devient Disponibilité limitée.

## Procédure

1. Cliquez sur **Projet, Liste de fonctions de projet, Définir la qualité de service**.
2. Développez les noeuds de l'arborescence pour afficher la qualité de service pour chaque fonction.
3. Pour remplacer la qualité de service, cliquez sur la flèche en regard de chaque fonction et sélectionnez l'indicateur de qualité de service dans la liste.
4. Après avoir modifié la qualité de service, vous pouvez ajouter des informations détaillées sur la fonction dans la zone de texte située à droite.  
Ces informations deviennent disponibles pour les utilisateurs et peuvent les aider à déterminer s'ils veulent utiliser cette fonction dans leurs rapports.  
**Astuce :** Cliquez sur **Retirer le remplacement** pour rétablir la valeur par défaut de la qualité de service.
5. Cliquez sur **OK**.

## Contrôle et optimisation des requêtes

Vous pouvez modifier les propriétés d'une source de données qui a été créée à l'aide de l'assistant de métadonnées dans IBM Cognos Framework Manager. Les propriétés de source de données vous permettent de contrôler et d'optimiser l'exécution des requêtes sur la base de données.

Vous ne pouvez pas modifier les propriétés d'une source de données qui a été créée à l'aide du portail. Ces sources de données peuvent uniquement être modifiées dans le portail. Pour plus d'informations, voir le document IBM Cognos *Administration and Security Guide*.

Propriété de source de données	Description
Nom	Nom descriptif de la connexion de source de données fourni par l'utilisateur au moment de la création.
Traitement de requête	Détermine si le traitement SQL est effectué par le serveur de base de données ou en local.
Traitement du cumul	Détermine si les cumuls d'agrégation sont calculés en local dans la base de données
Mode d'accès de la transaction	Définit le mode d'accès d'une nouvelle transaction. Pour plus d'informations, voir «Définition des modes de transaction», à la page 320.
Mode de l'instruction de transaction	Définit l'action qui suit la fin d'une transaction. Pour plus d'informations, voir «Définition des modes de transaction», à la page 320.
Source de données Content Manager	Indique le nom de la source de données, tel qu'il est identifié dans Content Manager. Si vous utilisez une source de données XML, cette propriété peut être paramétrée.
Catalogue	Représente différentes informations pour différentes bases de données. Par exemple, si la base de données est SQL Server, l'élément contient le nom de la base de données ; si la base de données est Oracle, il n'est pas utilisé.
cube	Indique le nom du cube
Schéma	Représente différentes informations pour différentes bases de données. Par exemple, pour SQL Server ou Oracle, l'élément contient le nom du propriétaire.
Type	Indique le type de l'objet parent.

Propriété de source de données	Description
Type de requête	Indique le type de modèle de requête compris par cette source de données. Par exemple, les sources SQL sont relationnelles et les sources MDS sont multidimensionnelles.
Interface de requête	Cet élément contient deux lettres qui identifient le type de fournisseur. Il est géré par l'application.
Identificateur d'ensemble de fonctions	Définit l'ensemble de fonctions qui s'applique à une source de données. Utilisé lors du remplissage initial des ensembles de fonctions d'une vue de sécurité lorsqu'un pack est créé.
Tables de mappage des paramètres	Fait référence à une table de mappage des paramètres qui représente une mappe de table des alias Oracle Essbase.





---

## Chapitre 9. Guide de modélisation des métadonnées

IBM Cognos Framework Manager est un outil de modélisation des données qui permet de générer des requêtes pour IBM Cognos BI. Un modèle est un ensemble de métadonnées comprenant des informations physiques et professionnelles pour une ou plusieurs sources de données. IBM Cognos BI permet la gestion des performances sur des sources de données relationnelles normalisées ou dénormalisées, ainsi que sur diverses sources de données OLAP.

Pour ouvrir le *Guide de modélisation des métadonnées IBM Cognos* dans une autre langue, accédez à `emplacement_installation\c10\webcontent\documentation` et ouvrez le dossier pour la langue souhaitée. Ouvrez ensuite le fichier `ug_best.pdf`.

---

### Présentation des concepts de modélisation d'IBM Cognos

Avant de commencer, vous devez maîtriser les concepts de modélisation fondamentaux d'IBM Cognos concernant les métadonnées de modélisation à utiliser dans le cadre de la génération de rapports professionnels et de l'analyse de données.

#### Concepts de modélisation relationnelle

Lors d'une modélisation dans IBM Cognos Framework Manager, il est important de comprendre que vous n'êtes pas obligé de concevoir votre source de données selon un schéma en étoile parfait. Les schémas en flocons et les autres formes de schémas normalisés sont tout aussi acceptables si votre source de données est optimisée pour fournir les performances requises pour l'application. Il est généralement recommandé de créer un modèle logique correspondant aux concepts du schéma en étoile. Cette condition, requise pour IBM Cognos Analysis Studio, a par ailleurs déjà fait ses preuves en tant que moyen d'organisation efficace des données pour vos utilisateurs.

Lorsque vous commencez à développer votre application avec une source de données complexe, il est conseillé de créer une vue simplifiée représentant la façon dont vos utilisateurs perçoivent l'activité, en la concevant à l'aide des consignes fournies dans le présent document, afin d'obtenir les requêtes et résultats attendus. Un modèle relationnel bien construit sert de base à votre application et vous fournit un point de départ fiable pour tirer parti des fonctionnalités dimensionnelles des logiciels IBM Cognos.

Si vous commencez avec une source de données de schéma en étoile, le modèle est plus simple à réaliser car les concepts utilisés dans la création d'un schéma en étoile se prêtent bien à la construction d'applications de requête et d'analyse. Les consignes fournies dans le présent document vous aideront à concevoir un modèle répondant aux besoins de votre application.

#### Cardinalité

Des relations existent entre deux sujets de requête. La cardinalité d'une relation est le nombre de lignes liées à chacun des deux sujets de requête. Les lignes sont liées par l'expression de la relation ; cette expression fait généralement référence aux clés primaires et étrangères des tables sous-jacentes.

Le logiciel IBM Cognos utilise la cardinalité d'une relation pour :

- éviter le double comptage des données de fait ;
- rendre en charge les jointures de boucles qui sont fréquentes dans les modèles à schéma en étoile ;
- optimiser l'accès au système de la source de données sous-jacente ;
- identifier les sujets de requête qui se comportent comme des faits ou dimensions.

Une requête qui utilise plusieurs faits de différentes tables sous-jacentes est divisée en requêtes distinctes pour chaque table de faits sous-jacente. Chaque requête de fait individuelle fait référence à sa table de faits correspondante ainsi qu'aux tables dimensionnelles liées à cette table de faits. Une autre requête est utilisée pour fusionner ces requêtes individuelles en ensemble unique de résultats. La requête de cette dernière opération est généralement appelée requête liée. Vous savez qu'il s'agit d'une requête liée lorsque vous pouvez voir coalesce ainsi qu'une jointure externe intégrale.

La requête liée permet également au logiciel IBM Cognos de mettre correctement des données en rapport à divers niveaux de granularité. Voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

#### **Cardinalité des requêtes générées :**

Le logiciel IBM Cognos prend en charge la cardinalité minimale/maximale et la cardinalité facultative.

Dans  $0:1$ ,  $0$  est la cardinalité minimale et  $1$  est la cardinalité maximale.

Dans  $1:n$ ,  $1$  est la cardinalité minimale et  $n$  est la cardinalité maximale.

Une relation dont la cardinalité est définie comme  $1:1$  à  $1:n$  est généralement considérée en tant que  $1$  à  $n$  lors de l'analyse des cardinalités maximales.

Une cardinalité minimale de  $0$  signifie que la relation est facultative. Vous pouvez définir une cardinalité minimale de  $0$  lorsque vous souhaitez que la requête conserve les informations de l'autre côté de la relation en l'absence de correspondance. Par exemple, une relation entre le client et les ventes réelles peut être définie comme  $1:1$  à  $0:n$ . Cela indique que les rapports vont afficher les informations relatives au client demandé même s'il n'y a aucune donnée sur les ventes.

Par conséquent, une relation  $1$  à  $n$  peut également être indiquée sous la forme suivante :

- $0:1$  à  $0:n$
- $0:1$  à  $1:n$
- $1:1$  à  $0:n$
- $1:1$  à  $1:n$

Utilisez l'instruction **Incidence de la relation** de la boîte de dialogue **Définition de relation** pour mieux comprendre la cardinalité. Par exemple, Personnel de vente ( $1:1$ ) est joint à Commandes ( $0:n$ ).

Relationship impact:	Each Order has one and only one Sales Staff. Each Sales Staff has zero or more Order (outer join).
----------------------	---

Il est important de vérifier que la cardinalité est correctement prise en compte dans le modèle, car elle détermine la détection des sujets de requête de faits et permet d'éviter le double comptage des données factuelles.

Le logiciel IBM Cognos utilise les règles de base suivantes pour appliquer la cardinalité lors de la génération des requêtes :

- La cardinalité est appliquée dans le contexte d'une requête.
- La cardinalité 1 à n implique des données de fait du côté n et des données de dimension du côté 1.
- Un sujet de requête peut se comporter en tant que sujet de requête de fait ou en tant que sujet de requête dimensionnel, selon les relations requises pour répondre à une requête particulière.

Utilisez **Model Advisor** pour afficher l'évaluation du comportement impliqué par la cardinalité dans votre modèle.

Pour plus d'informations, voir «Requête à fait unique», à la page 365 et «Requête à granularité multiple et à faits multiples sur des dimensions conformes», à la page 367.

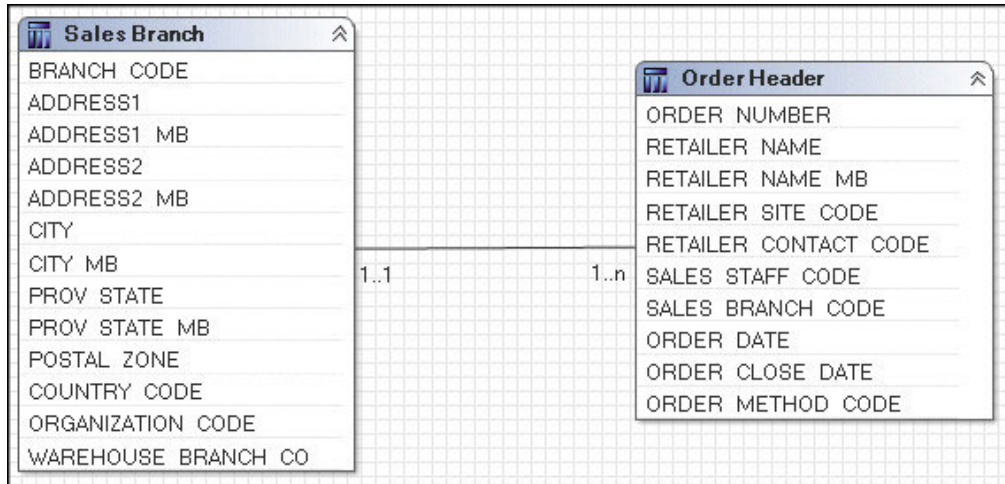
#### **Cardinalité dans le contexte d'une requête :**

Le rôle de la cardinalité dans le contexte d'une requête est essentiel car la cardinalité sert à déterminer où et quand diviser la requête lors de la génération de requêtes à faits multiples. Si les dimensions et les faits ne sont pas identifiés, des requêtes liées risquent d'être inutilement créées, ce qui pèse sur les performances, ou bien les requêtes peuvent être formées de façon incorrecte, ce qui peut produire des résultats incorrects.

Les exemples ci-dessous illustrent la façon dont la cardinalité est interprétée par le logiciel IBM Cognos.

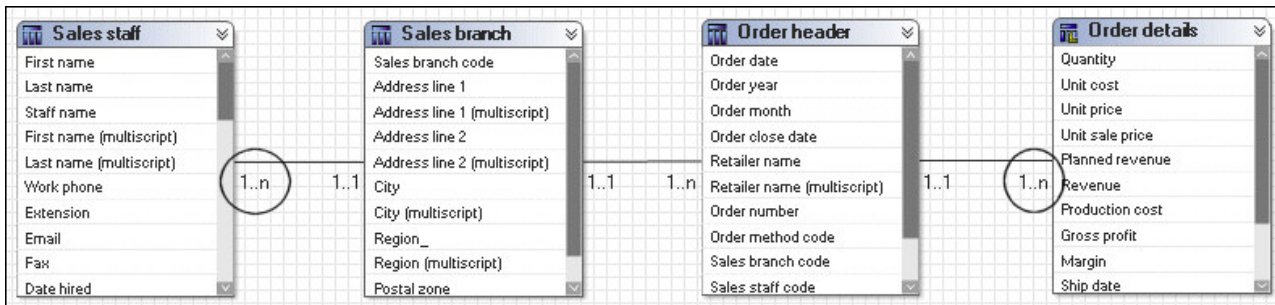
*Exemple : Sujets de requête qui se comportent comme une dimension et un fait :*

Dans cet exemple, le sujet de requête Succursale se comporte comme une dimension par rapport au sujet de requête En-tête de commande et ce dernier se comporte comme un fait par rapport au sujet de requête Succursale.



Exemple : Quatre sujets de requête inclus dans une requête :

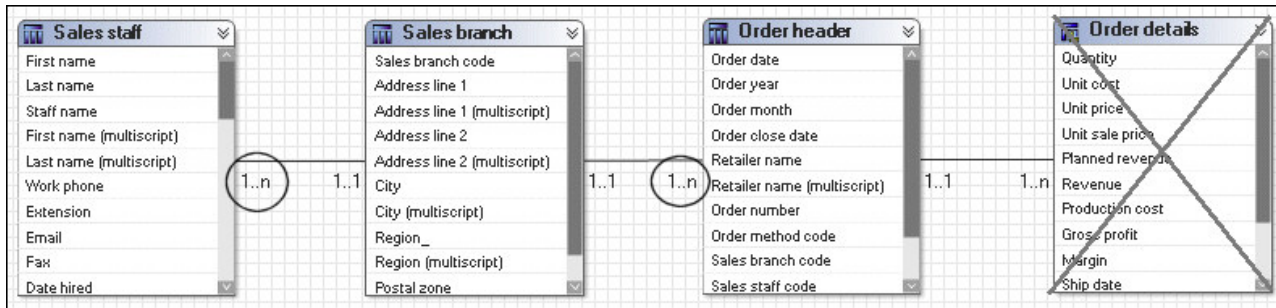
Dans l'exemple qui suit, les quatre sujets de requête sont inclus dans une requête. Personnel de vente et Détails de commande sont traités comme des faits. En-tête de commande et Succursale sont traités comme des dimensions.



Le code SQL généré pour cette requête sera divisé, il traitera Personnel de vente et Détails de commande comme des faits. Les résultats de ces deux sous-requêtes sont liés à l'aide des informations extraites du sujet de requête Succursale. Cela génère un rapport répertoriant les informations sur le personnel de vente par succursale à côté des détails sur les commandes et des en-têtes de commande par succursale.

Exemple : Trois sujets de requête inclus dans une requête :

Dans l'exemple qui suit, seuls trois sujets de requête sont inclus dans une requête. Le sujet de requête Détails de commande n'est pas utilisé. Le sujet de requête En-tête de commande est à présent traité comme un fait. Le sujet de requête Personnel de vente est toujours traité comme un fait.



Le code SQL de cet exemple a également généré une requête liée, qui renvoie un résultat similaire à celui obtenu précédemment. Notez qu'une opération liée conserve les informations des deux côtés de l'opération à l'aide d'une jointure externe complète.

## Déterminants

Les déterminants reflètent la granularité en représentant des sous-ensembles ou des groupes de données dans un sujet de requête et ils permettent de garantir l'agrégation correcte de ces données répétées. Les déterminants sont étroitement liés au concept de clés et d'index de la source de données et sont importés en fonction des informations de clé et d'index uniques contenues dans cette source de données. Il est conseillé de toujours vérifier les déterminants importés et, si nécessaire, de modifier ces derniers ou d'en créer d'autres. Lorsque vous modifiez les déterminants, vous pouvez redéfinir les informations de clé et d'index contenues dans votre source de données, en les remplaçant par des informations qui correspondent mieux à vos besoins en matière d'analyse et de génération de rapports. En ajoutant des déterminants, vous pouvez représenter des groupes de données répétées qui sont pertinentes pour votre application.

Voici un exemple de déterminant unique, Jour, dans l'exemple Temps ci-dessous. L'exemple de déterminant non unique est Mois ; la clé dans Mois est répétée selon le nombre de jours d'un mois donné. Lorsque vous définissez un déterminant non unique, vous devez spécifier **Grouper par**. Cela indique à IBM Cognos que lorsque les clés ou attributs associés à ce déterminant sont répétés dans les données, il convient d'appliquer des fonctions d'agrégation et un groupement afin d'éviter le double comptage. Il n'est pas conseillé de spécifier des déterminants en sélectionnant à la fois **Identifié de manière unique** et **Grouper par** ou aucune de ces deux valeurs.

Clé Année	Clé Mois	Nom du mois	Clé Jour	Nom du jour
2006	200601	6 janvier	20060101	Dimanche 1er janvier 2006
2006	200601	6 janvier	20060102	Lundi 2 janvier 2006

Vous pouvez définir trois déterminants pour ce fichier de la façon suivante : deux déterminants **Regrouper par** (Année et Mois) et un déterminant unique (Jour). Ce concept est similaire mais pas identique au concept de niveaux et de hiérarchies.

Nom du déterminant	Clé	Attributs	Identifié de manière unique	Grouper par
Année	Clé Année	Aucun	Non	Oui
Mois	Clé Mois	Nom du mois	Non	Oui
Jour	Clé Jour	Nom du jour Clé Mois Nom du mois Clé Année	Oui	Non

Dans ce cas, une seule clé est utilisée pour chaque déterminant car chaque clé contient suffisamment d'informations pour identifier un groupe parmi les données. Le sujet Mois est souvent difficile à gérer lorsque la clé ne contient pas assez d'informations pour indiquer clairement à quelle année le mois appartient. Dans ce cas, cependant, la clé Mois contient la clé Année qui à elle seule permet d'identifier les mois en tant que sous-groupe des années.

**Remarque :** Vous pouvez créer un déterminant qui regroupe les mois sans le contexte des années, mais ce choix est peu courant car le rapport correspondant contiendrait toutes les données du mois de février de toutes les années regroupées, au lieu des seules données de février 2006.

### Utilisation de déterminants avec des clés multiparties

Dans l'exemple de dimension de temps vu précédemment, une clé suffisait à identifier chaque ensemble de données pour un déterminant, mais ce n'est pas toujours le cas.

Par exemple, la dimension géographique ci-dessous utilise des définitions de clés multiparties pour tous les déterminants sauf un.

Région	Clé Region	Clé Etat/Province	Clé Ville
Amérique du Nord	USA	Illinois	Springfield
Amérique du Nord	USA	Missouri	Springfield
Amérique du Nord	USA	Californie	Dublin
Europe	Irlande	n.d.	Dublin

A l'instar de l'exemple Temps, vous pouvez définir trois déterminants pour ce fichier de la façon suivante : deux déterminants **Grouper par** (Région et Etat/Province) et un déterminant unique (Ville).

Nom du déterminant	Clé	Attributs	Identifié de manière unique	Grouper par
Région	Clé Region	Aucun	Non	Oui
Etat/Province	Clé Etat/Province	Aucun	Non	Oui

Nom du déterminant	Clé	Attributs	Identifié de manière unique	Grouper par
Ville	Clé Region Clé Etat/Province Clé Ville	Aucun	Oui	Non

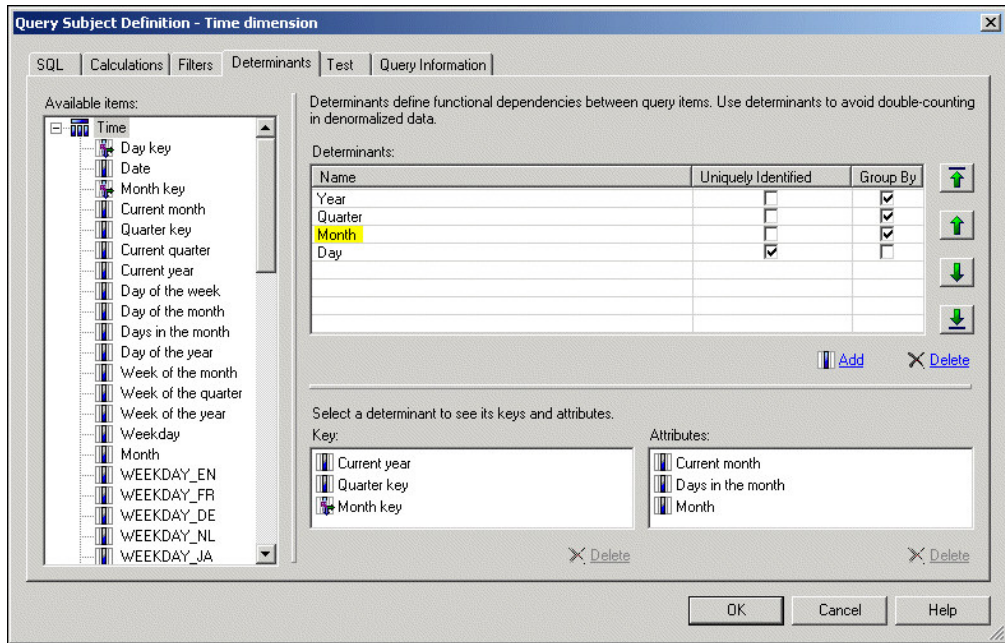
Dans ce cas, nous avons utilisé la clé Region, la clé Etat/Province et la clé Ville pour garantir l'unicité de Ville. Nous avons procédé ainsi car dans les données qui nous ont été fournies, certains noms de villes étaient communs à plusieurs états ou provinces, qui à leur tour étaient répétées pour les régions.

### **Les déterminants sont évalués dans l'ordre où ils sont indiqués.**

Il n'existe pas de concept d'une hiérarchie au sein des déterminants, mais un ordre d'évaluation s'applique. Lorsqu'IBM Cognos examine une sélection d'éléments à partir d'un sujet de requête, il les compare à chaque déterminant (clés et attributs) un par un, selon l'ordre défini dans l'onglet **Déterminants**. De cette façon, IBM Cognos sélectionne le déterminant qui correspond le mieux.

Dans l'exemple suivant, les attributs pour le mois en cours, les jours dans le mois et les noms de mois localisés sont associés à la clé Mois. Lorsque la requête soumise fait référence à l'un de ces attributs, le déterminant Mois est le premier déterminant sur lequel le critère de correspondance est satisfait. Si aucun autre attribut n'est requis, l'évaluation des déterminants s'arrête à Mois et ce déterminant est utilisé pour les clauses group et for dans le code SQL.

Lorsque d'autres attributs de la dimension sont également inclus, mais sans avoir été mis en correspondance avec un déterminant précédent, IBM Cognos continue son évaluation jusqu'à ce qu'une correspondance soit trouvée ou que le dernier déterminant soit atteint. C'est pour cette raison que tous les éléments de requête sont associés à un déterminant unique. Si aucune autre correspondance n'est trouvée, la clé unique de l'ensemble de données complet sert à déterminer la façon dont les données sont groupées.



## Situations dans lesquelles utiliser des déterminants

Les déterminants peuvent servir à résoudre différents problèmes liés à la granularité des données, vous devez les utiliser systématiquement dans les principaux cas suivants :

- Un sujet de requête qui se comporte comme une dimension a plusieurs niveaux de granularité et sera joint à des données de faits sur différents ensembles de clés.

Par exemple, le sujet Temps a plusieurs niveaux et il est joint à Inventaire sur la clé Mois et à Ventes sur la clé Jour. Pour plus d'informations, voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

- Il est nécessaire d'effectuer un comptage ou d'autres fonctions d'agrégation sur une clé ou un attribut qui est répété.

Par exemple, Temps a une clé Mois et un attribut, Nombre de jours dans le mois, qui est répété pour chaque jour. Si vous voulez utiliser l'attribut Nombre de jours dans le mois dans un rapport, vous ne souhaitez pas obtenir la somme des jours du mois pour chaque jour du mois. En revanche, vous voulez obtenir la valeur unique de Nombre de jours dans le mois pour la clé Mois choisie. Dans SQL, cela donne `XMIN(Days in the month for Month_Key)`. Il existe également une clause `Group by` dans le code SQL Cognos.

Il existe quelques cas moins fréquents dans lesquels il convient d'utiliser des déterminants :

- Vous souhaitez identifier de façon unique la ligne de données lors de l'extraction de données texte BLOB de la source de données.

L'utilisation de requêtes sur des données BLOB requiert des informations de clé ou d'index supplémentaires. Si ces informations ne sont pas disponibles dans la source de données, vous pouvez les ajouter à l'aide de déterminants. Remplacez les déterminants importés depuis la source de données qui sont en conflit avec des relations établies pour la génération de rapports.

Il n'est pas possible d'utiliser des clés à segments multiples lorsque le sujet de requête accède aux données BLOB. Avec des requêtes récapitulatives, les données BLOB doivent être extraites séparément de la partie récapitulative de la



requête. Pour cela, vous avez besoin d'une clé identifiant la ligne de façon unique et n'étant pas associée à plusieurs segments.

- Il existe une jointure définie qui utilise moins de clés qu'un déterminant unique défini pour un sujet de requête.

Si votre jointure est basée sur un sous-ensemble des colonnes qui sont référencées par les clés d'un déterminant unique du côté 0..1 ou 1..1 des relations, un conflit se produit. Pour résoudre ce conflit, modifiez la relation pour qu'elle soit parfaitement conforme au déterminant ou modifiez le déterminant de sorte qu'il prenne en charge la relation.

- Vous souhaitez remplacer les déterminants importés depuis la source de données qui sont en conflit avec des relations établies pour la génération de rapports.

Par exemple, il existe des déterminants pour deux sujets de requête correspondant à plusieurs colonnes, mais la relation entre ces sujets de requête n'utilise qu'un sous-ensemble de ces colonnes. Modifiez les informations des déterminants du sujet de requête si elles ne sont pas adaptées à l'utilisation de colonnes supplémentaires dans la relation.

### Requêtes à granularité multiple et à faits multiples

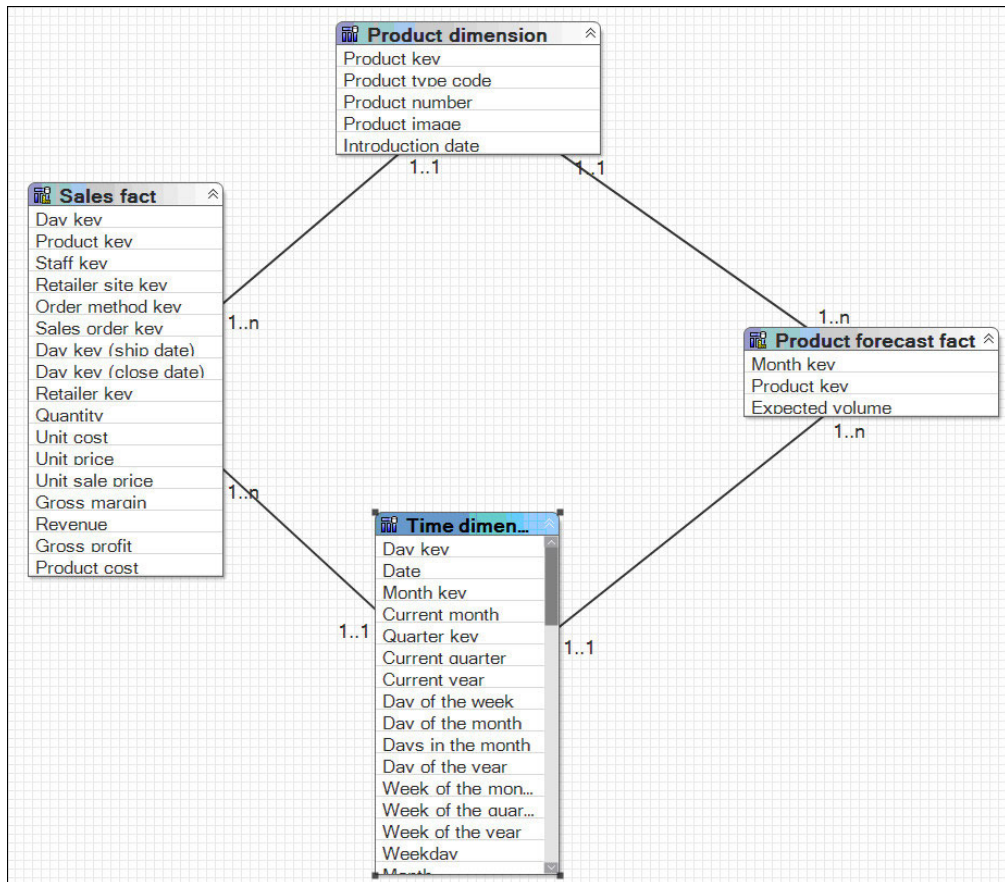
Des requêtes à granularité multiple et à faits multiples dans des sources de données relationnelles se produisent lorsqu'une table contenant des données dimensionnelles est jointe à plusieurs tables de faits sur différentes colonnes de clés.

Notez que dans la présente section le terme dimension est utilisé au sens conceptuel. Un sujet de requête avec une cardinalité de 1:1 or 0:1 se comporte comme une dimension. Pour plus d'informations, voir «Cardinalité», à la page 327.

Un sujet de requête comporte généralement des groupes ou niveaux distincts de données d'attributs avec des clés qui se répètent. Les studios IBM Cognos procèdent automatiquement à une agrégation au plus bas niveau commun de granularité dans le rapport. Le risque de double comptage apparaît lors de la création de totaux dans les colonnes contenant des données qui se répètent. Si le niveau de granularité des données est correctement modélisé, le double comptage peut être évité.

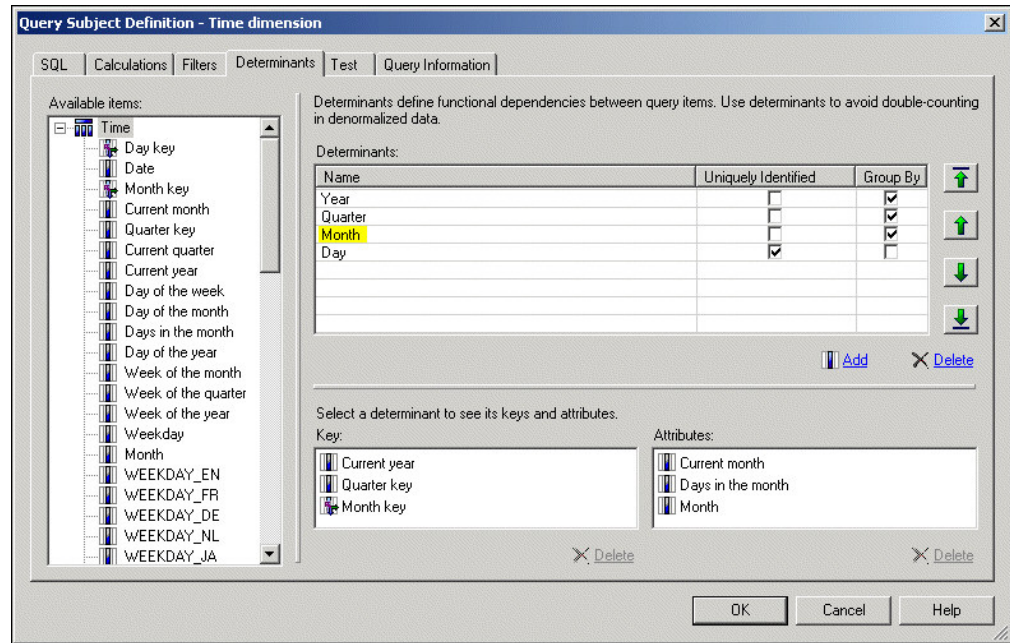
**Remarque :** il est possible de consigner des données à un niveau de granularité inférieur au niveau commun le plus bas. Dans ce cas, les données d'un niveau de granularité plus élevé se répètent, mais cela n'a pas d'incidence sur les totaux si les déterminants sont correctement appliqués.

L'exemple ci-dessous illustre deux sujets de requête de fait, Sales et Product forecast, qui partagent deux sujets de requête dimensionnels, Time et Product.



Dans cet exemple, la dimension Time représente le point central du problème de granularité. Le fait Sales est joint à la dimension Time sur la clé Day et le fait Product forecast est joint à la dimension Time sur la clé Month. En raison des différentes clés de jointure, au moins deux déterminants doivent être clairement identifiés sur la dimension Time. Par exemple, les déterminants des clés Month et Day sont identifiés. Day est la clé unique de la dimension Time, les clés Month sont répétées pour chaque jour du mois.

Par exemple, les déterminants de Month se présentent comme suit :



Le sujet de requête Product pourrait avoir au moins trois déterminants : Product line, Product type et Product. Il dispose de relations avec les deux tables de faits sur la clé Product. Il n'existe aucun problème de granularité au niveau du sujet de requête Product.

Par défaut, un rapport est agrégé pour extraire des enregistrements de chaque table de faits au niveau de granularité commun le plus bas. Si vous créez un rapport utilisant Quantité dans Sales, Volume prévu dans Product forecast, Mois dans Time et Nom de produit dans Product, le rapport extrait les enregistrements de chaque table de faits au niveau de granularité commun le plus bas. Dans l'exemple qui suit, il s'agit des niveaux Mois et Produit.

Pour éviter tout double comptage lorsqu'il existe des données à plusieurs niveaux de granularité, créez au moins deux déterminants pour le sujet de requête Time. Pour obtenir un exemple, voir «Déterminants», à la page 331.

Mois	Nom de produit	Quantité	Volume prévu
Avril 2007	Calmo brûlures	1 410	1 690
Avril 2007	Parapluie Omnium	132	125
Février 2007	Calmo brûlures	270	245
Février 2007	Parapluie Omnium		1
Février 2006	Calmo brûlures	88	92

Si vous n'indiquez pas correctement les déterminants dans le sujet de requête Time, l'agrégation risque de ne pas être correcte. Par exemple, les valeurs Volume prévu qui existent au niveau Mois dans Product forecast sont répétées pour chaque jour dans le sujet de requête Time. Lorsque les déterminants ne sont pas définis correctement, les valeurs Volume prévu sont multipliées par le nombre de jours dans le mois.

Mois	Nom de produit	Quantité	Volume prévu
Avril 2007	Calmo brûlures	1 410	50 700
Avril 2007	Parapluie Omnium	132	3 750
Février 2007	Calmo brûlures	270	7 134
Février 2007	Parapluie Omnium		29
Février 2006	Calmo brûlures	88	2 576

Notez les différents chiffres dans la colonne Volume prévu.

## Considérations sur la conception de modèles

Lors de la construction d'un modèle, il est important de comprendre qu'il n'existe pas de flux de travaux unique pouvant fournir un modèle adapté à toutes les applications. Avant de commencer votre modèle, vous devez connaître les spécifications de l'application en matière de fonctionnalité, facilité d'utilisation et performances. La conception de la source de données et les spécifications de l'application vont déterminer les réponses à de nombreuses questions dans cette section.

### Où les relations et déterminants doivent-ils être créés ?

L'endroit où créer les relations fait partie des questions fréquemment posées. Les relations doivent-elles être créées entre les sujets de requête de source de données, entre les sujets de requête de modèle ou entre les deux ? La réponse peut varier car elle dépend de la complexité de la source de données que vous êtes en train de modéliser.

Lorsque vous utilisez des sujets de requête de source de données, les relations et déterminants s'appartiennent mutuellement.

Lorsque vous utilisez des sujets de requête de modèle, le recours à des relations et à des déterminants a certains effets secondaires.

- Le sujet de requête de modèle commence à fonctionner comme une vue, ce qui a pour conséquence de remplacer le paramètre **Comme une vue** ou **Minimisée** du type **SQL Generation** pour un sujet de requête.

Cela signifie que le code SQL reste le même, quel que soit l'élément du sujet de requête auquel il est fait référence. Pour plus d'informations, voir «Définition du langage SQL réduit», à la page 339.

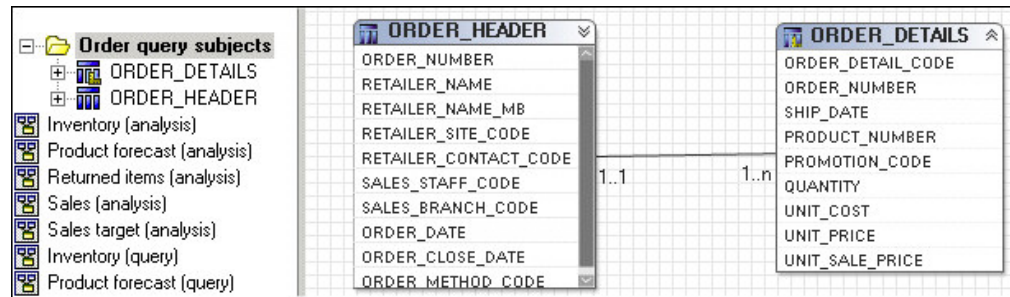
- Le sujet de requête de modèle devient un objet autonome.

Cela signifie que les relations sous-jacentes ne sont plus appliquées, à l'exception de celles qui existent entre les objets référencés. Il peut s'avérer nécessaire de créer des relations supplémentaires pour palier celles qui étaient précédemment induites par les métadonnées des sujets de requête sous-jacents.

- Lorsqu'un déterminant est créé sur un sujet de requête de modèle, ce déterminant est ignoré tant qu'une relation n'est pas créée avec lui.

Voici un exemple de relation sur un sujet de requête de modèle qui remplace le paramètre **SQL réduit** et simplifie le modèle. Dans cet exemple, En-têtes de commande et Détails de commande sont combinés de façon à se comporter comme un fait unique. Ils sont placés dans leur propre dossier et toutes leurs relations sont supprimées, à l'exception de la relation entre En-têtes de commande et Détails de commande. C'est la seule relation qui sera effective une fois que le sujet de requête

de modèle est créé et que des relations y sont rattachées.



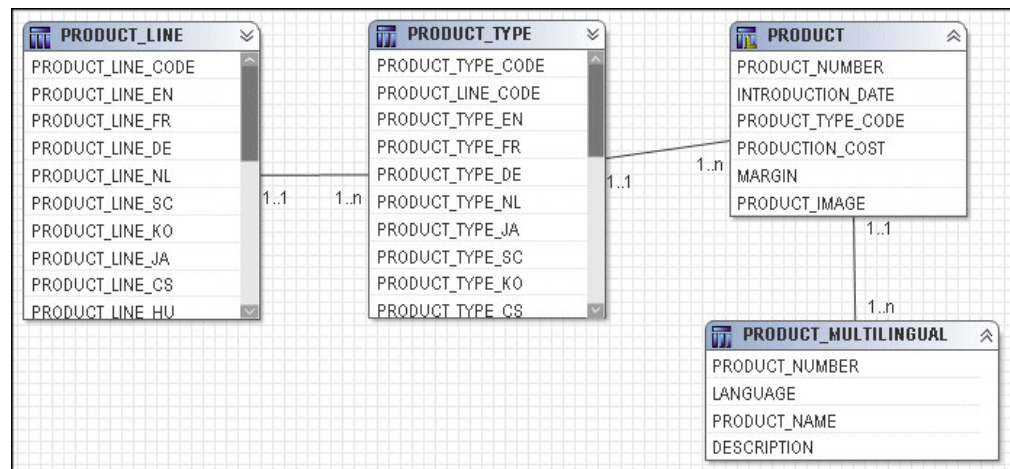
Pour choisir où définir les relations et déterminants dans le modèle, vous devez connaître l'impact du langage SQL réduit sur votre application.

Pour plus d'informations sur les relations, les déterminants et le SQL réduit, voir les rubriques sur **Model Advisor** dans le manuel *IBM Cognos Framework Manager User Guide*.

### Définition du langage SQL réduit

Lorsque vous utilisez le langage SQL réduit, le code SQL généré contient uniquement l'ensemble de tables et de jointures minimum nécessaire à l'obtention des valeurs pour les éléments de requête sélectionnés.

Pour comprendre la signification du SQL réduit, utilisez les tables Produit suivantes. Quatre sujets de requête, Ligne de produits, Type de produit, Produit et Produit multilingue, sont tous joints les uns aux autres.



Ils peuvent être combinés dans un sujet de requête de modèle.

Products (model)
PRODUCT_LINE_CODE
PRODUCT_LINE
PRODUCT_TYPE_CODE
PRODUCT_TYPE
PRODUCT_NUMBER
PRODUCT_NAME
DESCRIPTION
PRODUCTION_COST
MARGIN
INTRODUCTION_DATE
PRODUCT_IMAGE

Si vous testez le sujet de requête de modèle Produits dans son ensemble, vous pouvez voir que la clause from de la requête fait référence à quatre tables.

```
select
  PRODUCT_LINE.PRODUCT_LINE_CODE as Product_Line_Code,
  PRODUCT_LINE.PRODUCT_LINE_EN as Product_Line,
  PRODUCT_TYPE.PRODUCT_TYPE_CODE as Product_Type_Code,
  PRODUCT_TYPE.PRODUCT_TYPE_EN as Product_Type,
  PRODUCT.PRODUCT_NUMBER as Product_Number,
  PRODUCT_MULTILINGUAL.PRODUCT_NAME as Product_Name
  PRODUCT_MULTILINGUAL.DESCRPTION as Product_Description,
  PRODUCT.INTRODUCTION_DATE as Introduction_Date,
  PRODUCT.PRODUCT_IMAGE as Product_Image,
  PRODUCT.PRODUCTION_COST as Production_Cost,
  PRODUCT.MARGIN as Margin
from
  gos1_82..gos1.PRODUCT_LINE PRODUCT_LINE,
  gos1_82..gos1.PRODUCT_TYPE PRODUCT_TYPE,
  gos1_82..gos1.PRODUCT PRODUCT,
  gos1_82..gos1.PRODUCT_MULTILINGUAL PRODUCT_MULTILINGUAL
where
  (PRODUCT_MULTILINGUAL."LANGUAGE" = N'EN')
  and
  (PRODUCT_LINE.PRODUCT_LINE_CODE = PRODUCT_TYPE.PRODUCT_LINE_CODE)
  and
  (PRODUCT_TYPE.PRODUCT_TYPE_CODE = PRODUCT.PRODUCT_TYPE_CODE)
  and
  (PRODUCT.PRODUCT_NUMBER = PRODUCT_MULTILINGUAL.PRODUCT_NUMBER
```

Si vous testez uniquement Nom de produit, vous pouvez constater que la requête qui en résulte utilise seulement Produit multilingue, qui est la table requise. Il s'agit des effets du SQL réduit.

```
select
  PRODUCT_MULTILINGUAL.PRODUCT_NAME as Product_Name
from
  gos1_82..gos1.PRODUCT_MULTILINGUAL PRODUCT_MULTILINGUAL
where
  (PRODUCT_MULTILINGUAL."LANGUAGE" = N'EN')
```

### Exemple: Lorsque le SQL réduit est important

Si vous modélisez une source de données normalisée, pensez au SQL réduit car il diminuera le nombre de tables utilisées dans certaines demandes et les performances s'en trouveront améliorées. Dans ce cas, il est préférable de créer des

relations et des déterminants entre les sujets de requête de la source de données, puis de créer des sujets de requête de modèle qui n'ont pas de relation.

Certains croient à tort qu'en l'absence de relation entre les objets, il est impossible de créer des groupes sous forme de schémas en étoile. Cela est faux. Sélectionnez les sujets de requête de modèle à inclure dans le groupe et utilisez l'assistant **Créer un regroupement de schémas en étoile**. Vous pouvez également créer des raccourcis et les déplacer dans un nouvel espace-noms. Les raccourcis vers les relations ne sont pas nécessaires ; cette fonctionnalité est purement visuelle dans le diagramme. L'effet est le même sur la génération des requêtes et la présentation dans les studios.

### **Exemple : Lorsque le SQL réduit n'est pas aussi important que les requêtes prévisibles**

Certains éléments d'une source de données doivent être encapsulés pour garantir qu'ils se comportent comme un objet de données unique. Il peut s'agir, par exemple, d'une table de sécurité qui doit être systématiquement jointe à un fait. Dans le modèle Ventes Vacances et Aventure, En-tête de commande et Détails de commande constituent un jeu de tables représentant un fait et doivent toujours faire l'objet d'une requête conjointe. Pour obtenir un exemple, voir «Où les relations et déterminants doivent-ils être créés ?», à la page 338.

### **Présentation de la mise en cache des métadonnées**

IBM Cognos Framework Manager stocke les métadonnées importées de la source de données. Cependant, selon les paramètres de restriction et certaines actions que vous effectuez dans le modèle, ces métadonnées peuvent ne pas être utilisées lors de la préparation d'une requête.

Si vous activez la restriction **Autoriser la portabilité du modèle amélioré lors de l'exécution**, Framework Manager interroge systématiquement la source de données pour obtenir des informations sur les métadonnées avant de préparer une requête. Si vous n'avez pas activé cette restriction, Framework Manager va, dans la plupart des cas, accéder aux métadonnées qui ont été stockées dans le modèle au lieu d'interroger la source de données. Les principales exceptions sont les suivantes :

- Le code SQL d'un sujet de requête de source de données a été modifié. Cela inclut l'utilisation de macros.
- Un calcul ou filtre a été ajouté à un sujet de requête de source de données.

**Remarque :** les requêtes de métadonnées générées sont bien prises en charge par la plupart des fournisseurs de systèmes de gestion de bases de données relationnelles et ne devraient donc pas avoir une incidence notable sur la majeure partie des applications de génération de rapports.

### **Sujets de requête et dimensions**

Les sujets de requête et les dimensions n'ont pas les mêmes objectifs. Le sujet de requête sert à générer des requêtes relationnelles et peut être créé à l'aide des règles du schéma en étoile, tandis que la dimension sert à la modélisation dimensionnelle des sources relationnelles, ce qui implique un comportement OLAP. Puisque les sujets de requête représentent la base des dimensions, la clé de la réussite pour tout modèle dimensionnel est un modèle relationnel bien construit.

Un modèle dimensionnel n'est nécessaire que si vous voulez utiliser IBM Cognos Analysis Studio afin d'activer les passages aux niveaux supérieurs et inférieurs dans les rapports, ou pour accéder aux fonctions membres dans les studios. Pour bon nombre d'applications, la fonctionnalité OLAP n'est pas nécessaire. Par

exemple, votre application est principalement destinée à la génération de rapports ou de requêtes ad hoc, sans qu'il soit nécessaire de passer au niveau supérieur ou inférieur, ou bien vous effectuez la maintenance d'un modèle IBM Cognos ReportNet. Dans les deux cas, vous pouvez publier des packs basés uniquement sur des sujets de requête.

Les déterminants des sujets de requête sont à distinguer des niveaux et des hiérarchies des dimensions ordinaires, mais ils peuvent être étroitement liés à une hiérarchie unique. Si vous envisagez d'utiliser vos sujets de requête en tant que base de vos dimensions, vous devez réfléchir à la structure des hiérarchies que vous souhaitez créer et vous assurer que vous avez créé des déterminants qui vont produire des résultats corrects lors de l'agrégation. Vérifiez que vous disposez des éléments suivants :

- Le sujet de requête doit avoir un déterminant défini pour chaque niveau de la hiérarchie dans la dimension ordinaire.
- Les déterminants doivent être définis dans le même ordre que les niveaux dans la dimension ordinaire.
- Si vous souhaitez plusieurs hiérarchies qui s'agrègent différemment, vous pouvez créer un sujet de requête supplémentaire avec d'autres déterminants en tant que source de l'autre hiérarchie.

La création d'un modèle relationnel complet qui fournit des résultats corrects et de bonnes performances vous garantit une bonne base pour développer un modèle dimensionnel. En outre, en faisant en sorte qu'une couche d'objets de modèle (des sujets de requête ou des dimensions) existe entre la source de données et les objets visibles dans les studios, il vous est plus facile de protéger vos utilisateurs des modifications.

### **Objets de modèle et raccourcis**

La principale différence entre les objets de modèle et les raccourcis est que les objets de modèle vous laissent la liberté d'inclure ou d'exclure des éléments et de les renommer. Vous pouvez utiliser des objets de modèle au lieu de raccourcis si vous devez limiter les éléments de requête à inclure ou renommer les éléments.

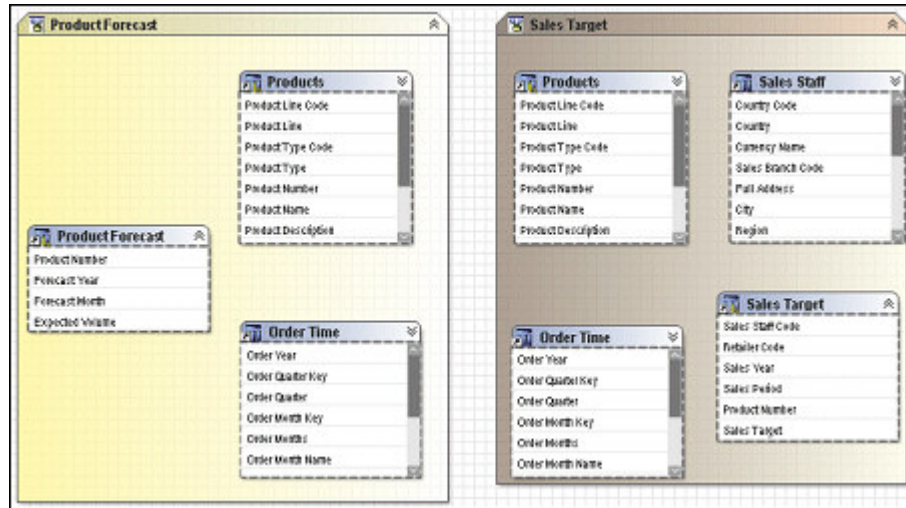
Les raccourcis sont moins souples que les objets de modèle du point de vue de la présentation, mais leur maintenance est beaucoup plus légère car ils sont automatiquement mis à jour dès que l'objet cible est mis à jour. Si la maintenance est l'une de vos principales préoccupations et que vous n'avez pas besoin de personnaliser l'apparence du sujet de requête, utilisez des raccourcis.

IBM Cognos Framework Manager dispose de deux types de raccourci :

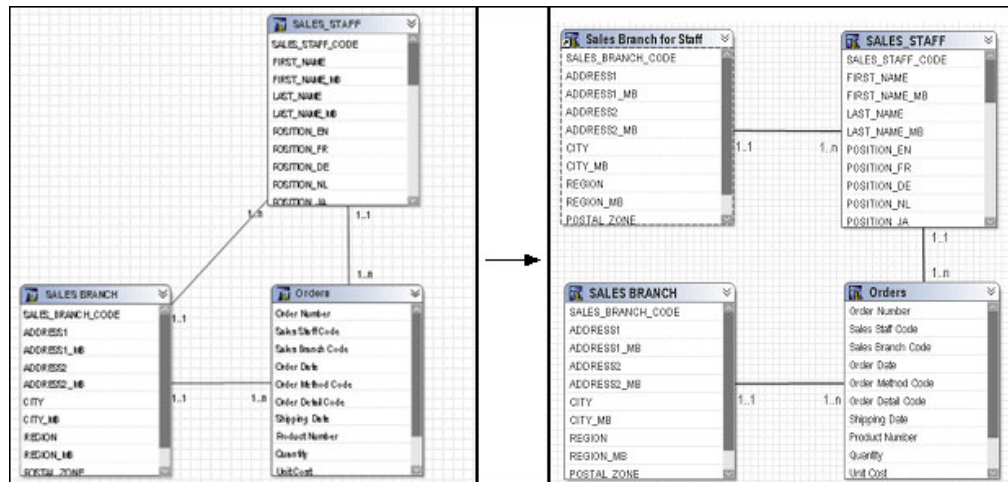
- les raccourcis ordinaires, qui sont une simple référence à l'objet cible ;
- les raccourcis d'alias, qui se comportent comme s'ils étaient une copie de l'objet d'origine avec un comportement complètement indépendant. Les raccourcis d'alias ne sont disponibles que pour les sujets de requête et les dimensions.

Les raccourcis ordinaires sont généralement utilisés en tant que dimensions conformes avec groupes sous forme de schémas en étoile, en créant plusieurs références avec exactement les mêmes nom et apparence à plusieurs endroits. Dans l'exemple ci-dessous, les raccourcis créés pour Produits et Heure de commande se comportent comme des références. Si une requête est écrite pour obtenir Produits à partir de Prévisions de vente et Objectifs de vente, la requête utilise la définition de Produits basée sur le raccourci d'origine et cette définition figure une seule fois dans la requête.





Les raccourcis d'alias sont généralement utilisés dans les dimensions de rôle ou les tables partagées. Il y a déjà un exemple relatif aux dimensions de rôle dans le présent document, nous allons donc étudier le cas des tables partagées. Dans cet exemple, Personnel de vente et Succursale peuvent être considérés comme des hiérarchies distinctes. Notre connaissance des données nous permet de savoir que les membres du personnel peuvent changer de succursale, il faut donc que nous puissions produire des rapports sur Succursale et Personnel de vente à la fois indépendamment et de façon groupée. Pour ce faire, nous devons créer un alias de Succursale pouvant être utilisé en tant que niveau dans la hiérarchie Personnel de vente.



Une fois le nouveau raccourci d'alias en place, il est possible de créer des requêtes qui demandent les commandes par succursale et les commandes par équipe commerciale, avec en même temps les informations relatives à leur succursale actuelle.

Lorsque vous ouvrez un modèle d'une édition précédente, la restriction **Traitement des raccourcis** a pour valeur **Automatique**. Lorsque l'option **Automatique** est utilisée, les raccourcis fonctionnent de la même façon que dans les éditions précédentes, c'est-à-dire qu'un raccourci qui se trouve dans le même répertoire que sa cible se comporte comme un alias, ou une instance indépendante, alors qu'un raccourci existant ailleurs dans le modèle se comporte comme une référence au

raccourci d'origine. Pour tirer parti de la propriété **Traiter en tant que**, il est conseillé de vérifier le modèle, et au moment de la réparation, d'affecter la valeur **Jointure explicite** à la restriction. La réparation modifie tous les raccourcis en leur affectant la valeur correcte à partir de la propriété **Traiter en tant que** selon les règles suivies par le paramètre **Automatique**. Cela signifie que le comportement de votre modèle ne changera pas tant que vous ne modifierez pas la propriété **Traiter en tant que** de vos raccourcis.

Lorsque vous créez un nouveau modèle, la valeur de la restriction **Traitement des raccourcis** est toujours **Jointure explicite**.

Lorsque la restriction a pour valeur **Jointure explicite**, le comportement du raccourci est dicté par la propriété **Traiter en tant que** et vous avez un contrôle total sur la façon dont les raccourcis se comportent sans avoir à connaître leur emplacement dans le modèle.

## Dossiers et espaces-noms

L'information la plus importante à connaître à propos des espaces-noms est qu'une fois la création du rapport commencée, tout changement de nom d'un espace-noms publié aura une incidence sur votre contenu IBM Cognos. En effet, le changement du nom de l'espace-noms modifie les identificateurs des objets publiés dans les métadonnées. Etant donné que l'espace-noms fait partie de l'identificateur d'objet dans IBM Cognos Framework Manager, chaque espace-noms doit avoir un nom unique dans le modèle. Chaque objet de l'espace-noms doit également avoir un nom unique. Une partie de la stratégie d'élaboration des groupes sous forme de schémas en étoile est de placer les raccourcis dans un espace-noms distinct, ce qui crée automatiquement un identificateur unique pour chaque objet dans l'espace-noms. Pour les bases de données relationnelles, cela vous permet d'utiliser le même nom pour les raccourcis vers des dimensions conformes dans différents groupes sous forme de schémas en étoile.

Lors de la prochaine tentative d'exécution d'une requête, d'un rapport ou d'une analyse sur le modèle mis à jour, vous obtenez une erreur. Si vous devez renommer l'espace-noms que vous avez publié, utilisez la commande **Analyze Publish Impact** pour identifier les rapports sur lesquels cela aura une incidence.

Les dossiers sont beaucoup moins compliqués que les espaces-noms. Ils n'ont qu'une fonction purement organisationnelle et n'ont aucune incidence sur l'identificateur des objets ou sur votre contenu. Vous pouvez créer des dossiers pour organiser les objets par sujet ou domaine fonctionnel. Cela facilite l'accès aux métadonnées, en particulier dans les projets volumineux.

Le principal inconvénient des dossiers réside dans le fait qu'ils nécessitent des noms uniques pour tous les sujets de requête, dimensions et raccourcis. Ils ne représentent donc pas la solution idéale pour contenir des objets partagés.

## Définition de l'ordre des opérations pour les calculs de modèles

Dans certains cas (notamment pour les calculs liés à un ratio), il est utile de procéder à l'agrégation sur les termes du calcul avant d'effectuer l'opération mathématique.

Par exemple, le fait Détails de commande suivant contient des informations sur chaque commande :

Order details	
Order detail code	
Order number	
Ship date	
Product number	
Quantity	
Unit cost	
Unit price	
Unit sale price	
Revenue	→ Regular Aggregate: Sum
Product cost	→ Regular Aggregate: Sum
Gross profit	
Margin	→ Regular Aggregate: Sum

L'élément Marge calcule la proportion de bénéfices :

$$\text{Margin} = (\text{Revenue} - \text{Product cost}) / \text{Revenue}$$

Si vous exécutez une requête pour afficher les revenus, le coût de produit et la marge de chaque produit en utilisant le fait Détails de commande, vous obtenez les résultats suivants :

Numéro de produit	Revenus	Coût de produit	Marge
1	\$23,057,141	\$11,292,005	61038 %
2	\$11,333,518	\$6,607,904	49606 %

Notez que la valeur de marge semble être erronée. Cela est dû à l'ordre des opérations utilisées dans le calcul de la marge. La marge est calculée de la façon suivante :

$$\text{Margin} = \text{sum} ( (\text{Revenue} - \text{Product cost}) / \text{Revenue} )$$

L'agrégation a eu lieu après l'opération mathématique et, dans ce cas, les résultats obtenus sont erronés.

Pour obtenir les résultats souhaités pour la marge, vous devez effectuer l'agrégation avant l'opération mathématique :

$$\text{Margin} = ( \text{sum}(\text{Revenue}) - \text{sum}(\text{Product cost}) ) / \text{sum}(\text{Revenue})$$

Vous obtenez les résultats suivants :

Numéro de produit	Revenus	Coût de produit	Marge
1	\$23,057.141	\$11,292,005	51,03 %
2	\$11,333,518	\$6,607,904	41,70 %

Cette opération peut être réalisée dans IBM Cognos Framework Manager en créant un calcul autonome pour la marge et en affectant la valeur **Calculé** à sa propriété **Agrégat ordinaire**. Chaque élément de requête figurant dans l'expression du calcul est agrégé comme indiqué dans la propriété **Agrégat ordinaire**. Les propriétés

**Agrégat ordinaire** pour le revenu et le coût de produit ont pour valeur **Sum**, par conséquent, lors du traitement du calcul, la fonction Sum est utilisée pour agréger ces termes.

**Remarque :** Le type d'agrégation calculée n'est pas pris en charge pour les calculs qui sont imbriqués dans des sujets de requête. Il est uniquement pris en charge pour les calculs autonomes et pour ceux qui sont imbriqués dans des dimensions de mesure et basés sur des mesures issues de la même dimension de mesure.

Examinons par exemple le calcul de marge imbriqué dans la dimension de mesure Ventes :

	Measures	Source
Revenue	Revenue	Model query subjects (gosales).Sales.Revenue
Product cost	Product cost	Model query subjects (gosales).Sales.Product cost
Margin	Margin	( Sales.Revenue - Sales.Product cost ) / Sales.Revenue

Dans cet exemple, la marge est basée sur les mesures Coût de produit et Revenu, qui font partie de la même dimension de mesure, Ventes. Si la propriété **Agrégat ordinaire** de la marge a pour valeur **Calculé**, la marge est cumulée de la façon suivante :

$$\text{Margin} = \text{sum}(\text{Revenue} - \text{Product cost} ) / \text{sum}(\text{Revenue})$$

Si la marge repose sur les éléments de requête source des mesures Coût de produit et Revenu (Sales (model).Product cost, Sales (model).Revenue), l'agrégation calculée n'est pas prise en charge et le comportement d'agrégation automatique est utilisé. Dans ce cas, la marge est cumulée de la façon suivante :

$$\text{Margin} = \text{sum}( \text{Revenue} - \text{Product cost} ) / \text{Revenue}$$

Pour plus d'informations sur la modification du mode d'agrégation des éléments de requête, voir le manuel *IBM Cognos Framework Manager User Guide*.

### Impact de la taille de modèle

La taille de votre modèle peut avoir une incidence sur l'efficacité de l'application Framework Manager.

Les modèles particulièrement volumineux entraînent des temps de traitement plus longs et, dans les cas extrêmes, des conditions de mémoire insuffisante. Les commandes telles que Analyser l'impact de publication, Rechercher des dépendances de rapport, Publier les packs et Exécuter Model Advisor s'exécutent de façon optimale sur des modèles de moins de 50 mégaoctets.

## Concepts de modélisation dimensionnelle

Les dimensions ordinaires et de mesure permettent l'activation d'une présentation OLAP des métadonnées, les passages au niveau supérieur et inférieur ainsi qu'une variété de fonctions OLAP. Vous devez utiliser des groupes sous forme de schémas en étoile (un fait avec plusieurs dimensions) pour utiliser Analysis Studio d'IBM Cognos Analysis Studio avec une source de données relationnelle.

Lors de la construction du modèle, il est conseillé de créer les dimensions ordinaires de modèle et les dimensions de mesure de modèle sur la base d'un modèle relationnel dans lequel des concepts de schéma en étoile ont été appliqués.

Vous pouvez convertir des sujets de requête de source de données en dimensions de source de données, mais ces dernières ont des fonctionnalités limitées par

rapport aux sujets de requête ou dimensions de modèle et ne sont pas recommandées pour une utilisation générale.

## Dimensions ordinaires

Les dimensions ordinaires représentent des données descriptives qui fournissent du contexte aux données modélisées dans des dimensions de mesure. Une dimension ordinaire est divisée en groupes d'informations appelés niveaux. A leur tour, les différents niveaux peuvent être organisés en hiérarchies. Par exemple, une dimension de produit peut contenir les niveaux Ligne de produits, Type de produit et Produit dans une hiérarchie unique appelée Produit. Autre exemple, une dimension de temps comprenant les niveaux Année, Trimestre, Mois, Semaine et Jour organisés en deux hiérarchies. La première hiérarchie, ATMJ, contient les niveaux Année, Trimestre, mois et Jour, tandis que la seconde hiérarchie AMJ contient les niveaux Année, Semaine et Jour.

La définition la plus simple d'un niveau est une clé métier et une légende, chacun d'eux faisant référence à un élément de requête. Une instance (ou ligne) d'un niveau est définie comme membre de ce niveau. Elle est identifiée par un nom unique de membre, lequel contient les valeurs des clés métier du niveau en cours et des niveaux supérieurs. Par exemple, [gosaies].[Products].[ProductsOrg].[Product]->[All Products].[1].[1].[2] identifie un membre qui se trouve au quatrième niveau, Product, de la hiérarchie ProductsOrg de la dimension [Products] qui est dans l'espace-noms [gosaies]. La légende de ce produit est Gamelle ChefDeCamp, qui est le nom affiché dans l'arborescence de métadonnées et dans le rapport.

Le niveau peut être défini comme unique si la clé métier du niveau est suffisante pour identifier chaque ensemble de données pour un niveau. Dans le modèle Ventes Vacances et Aventure, la définition de l'élément Type de produit n'est pas nécessaire pour les membres du niveau Produit car aucun numéro de produit n'est affecté à différents types de produit. Un niveau qui n'est pas défini comme unique est similaire à un déterminant qui utilise des clés multiparties, car des clés de niveaux de granularité supérieures sont nécessaires. Voir «Utilisation de déterminants avec des clés multiparties», à la page 332. Si des membres ne sont pas uniques au sein de membres ancêtres mais que le niveau est défini comme unique, les données des membres non uniques font l'objet d'un rapport sous la forme d'un seul membre. Par exemple, si le membre Ville est défini comme unique et identifié par un nom, les données relatives à Londres en Angleterre et à Londres au Canada seront combinées.

## Dimensions de mesure

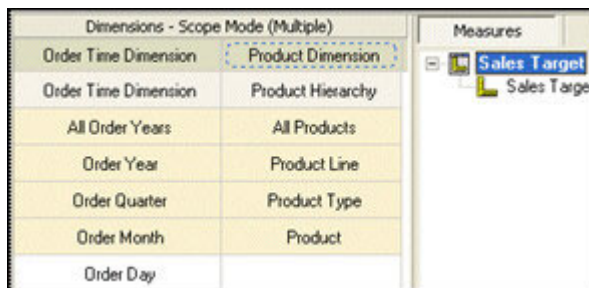
Les dimensions de mesure représentent les données quantitatives décrites par des dimensions ordinaires. Connue sous plusieurs noms dans différents produits OLAP, la dimension de mesure est tout simplement l'objet contenant les données de fait. Les dimensions de mesure diffèrent des sujets de requête de fait car elles n'incluent pas les clés étrangères utilisées pour joindre un sujet de requête de fait à un sujet de requête dimensionnel. En effet, la dimension de mesure n'est pas destinée à être jointe comme s'il s'agissait d'un objet de données relationnel. Pour la génération des requêtes, la dimension de mesure obtient sa relation à partir d'une dimension ordinaire par le biais des sujets de requête sous-jacents. De même, la relation vis-à-vis des autres dimensions de mesure s'effectue pour des dimensions ordinaires basées sur des sujets de requête construits de façon à se comporter comme des dimensions conformes. Pour activer les requêtes à granularité multiple et à faits multiples, vous devez disposer de sujets de requête et de déterminants créés de façon appropriée avant de construire des dimensions ordinaires et des dimensions de mesure.

## Relations de portée

Les relations de portée sont établies uniquement entre des dimensions de mesure et des dimensions ordinaires pour définir le niveau auquel les mesures sont disponibles à des fins de génération de rapports. Elles sont différentes des jointures et n'ont aucune incidence sur la clause Where. Dans une relation de portée, aucune condition ni critère ne sont définis sur la façon de former une requête. Elle indique uniquement si une dimension peut faire l'objet de requêtes avec un fait spécifique. L'absence de relation de portée peut entraîner une erreur lors de l'exécution ou provoquer un cumul des données de fait à un niveau supérieur à celui prévu, en raison des autres éléments du rapport.

Si vous définissez la relation de portée pour la dimension de mesure, les mêmes paramètres s'appliquent à toutes les mesures de la dimension de mesure. Si les données font l'objet de rapports à un niveau différent des mesures dans la dimension de mesure, vous pouvez définir la portée d'une mesure. Vous pouvez définir le niveau le plus bas à partir duquel les données peuvent faire l'objet d'un rapport.

Dans cet exemple, la dimension de mesure Objectifs de vente a une seule mesure comprenant le niveau Mois de la commande dans la dimension Heure de commande et le niveau Produit de la dimension Produit. Cela signifie que, lorsque vos utilisateurs tenteront d'accéder au détail au-delà du niveau des mois, ils verront des données répétées.



Dimensions - Scope Mode (Multiple)		Measures
Order Time Dimension	Product Dimension	Sales Target
Order Time Dimension	Product Hierarchy	Sales Target
All Order Years	All Products	
Order Year	Product Line	
Order Quarter	Product Type	
Order Month	Product	
Order Day		

---

## Construction du modèle relationnel

La modélisation dimensionnelle de sources de données relationnelles est possible dans IBM Cognos Framework Manager, mais elle requiert un modèle relationnel bien construit.

IBM Cognos ReportNet offrait des fonctionnalités dimensionnelles permettant la création de requêtes à faits multiples et empêchant le double comptage. Succédant à IBM Cognos ReportNet, le logiciel IBM Cognos offre des fonctionnalités conçues explicitement pour la représentation dimensionnelle des métadonnées et des capacités OLAP avec les sources de données relationnelles. Les concepts appliqués à la modélisation relationnelle dans IBM Cognos ReportNet sont maintenus même s'ils ont fait l'objet de quelques modifications, documentées dans le manuel *Framework Manager User Guide*.

Lorsque vous créez un nouveau modèle Framework Manager, vous devez suivre une procédure commune pour définir la génération de requêtes, même si vous n'avez pas l'intention d'utiliser les fonctionnalités de modélisation dimensionnelle. Vous devez modéliser de façon dimensionnelle une source de données relationnelle si vous voulez l'utiliser dans IBM Cognos Analysis Studio, afin d'activer les passages aux niveaux supérieurs et inférieurs dans les rapports, ou pour accéder aux fonctions membres dans les studios.

## Définition de la base de la modélisation relationnelle

Un modèle est l'ensemble des objets connexes requis par une ou plusieurs applications de génération de rapports associées. Un modèle relationnel bien construit est la base de tout modèle dimensionnel.

Lorsque vous définissez la base du modèle relationnel, procédez comme suit :

- Importez les métadonnées. Pour plus d'informations sur l'importation, voir le document *IBM Cognos Framework Manager User Guide*.
- «Vérification des métadonnées importées».
- «Résolution des relations ambiguës», à la page 350.
- Simplifiez le modèle relationnel à l'aide de concepts de schémas en étoile en analysant la cardinalité des faits et des dimensions et en choisissant l'emplacement des relations et des déterminants «Considérations sur la conception de modèles», à la page 338.
- Ajoutez des données de sécurité, si nécessaire. Pour plus d'informations sur la sécurité des données, voir le document *Framework Manager User Guide*.

Vous pouvez alors définir la représentation dimensionnelle du modèle, si nécessaire, et organiser le modèle pour la présentation.

### Vérification des métadonnées importées

Après avoir importé des métadonnées, vous devez les vérifier.

Vérifiez les points suivants :

- relations et cardinalité ;
- déterminants ;
- propriété **Utilisation** des éléments de requête ;
- propriété **Agrégat ordinaire** des éléments de requête.

Les notions de relation et de cardinalité sont présentées ici. Pour plus d'informations sur les propriétés **Utilisation** et **Agrégat ordinaire**, voir le manuel *Framework Manager User Guide*.

### Analyse de la cardinalité des faits et dimensions :

La cardinalité d'une relation définit le nombre de lignes d'une table liées aux lignes d'une autre table sur la base d'un ensemble (ou jointure) spécifique de clés. Dans IBM Cognos, la cardinalité permet d'identifier les sujets de requête qui se comportent comme des faits ou des dimensions. Il en résulte qu'IBM Cognos peut automatiquement résoudre une forme commune de jointure de boucle causée par des données de schéma en étoile, lorsque plusieurs tables de faits sont jointes à un ensemble commun de tables de dimensions.

Pour garantir une génération prévisible de requêtes, vous devez comprendre la façon dont la cardinalité est utilisée afin de l'appliquer correctement dans votre modèle. Il est conseillé d'examiner le schéma de la source de données sous-jacente et de modifier les zones où la cardinalité identifie de façon incorrecte les faits ou dimensions et risque de provoquer ainsi une génération imprévisible des requêtes. Vous pouvez utiliser la fonction **Model Advisor** de Framework Manager pour comprendre la manière dont la cardinalité est interprétée.

Pour plus d'informations, voir «Cardinalité», à la page 327.

## Résolution des relations ambiguës

Les relations ambiguës se produisent lorsque les données représentées par un sujet de requête ou une dimension sont visibles dans au moins deux contextes ou rôles ou qu'elles peuvent être jointes de plusieurs façons.

Les relations ambiguës les plus fréquentes sont les suivantes :

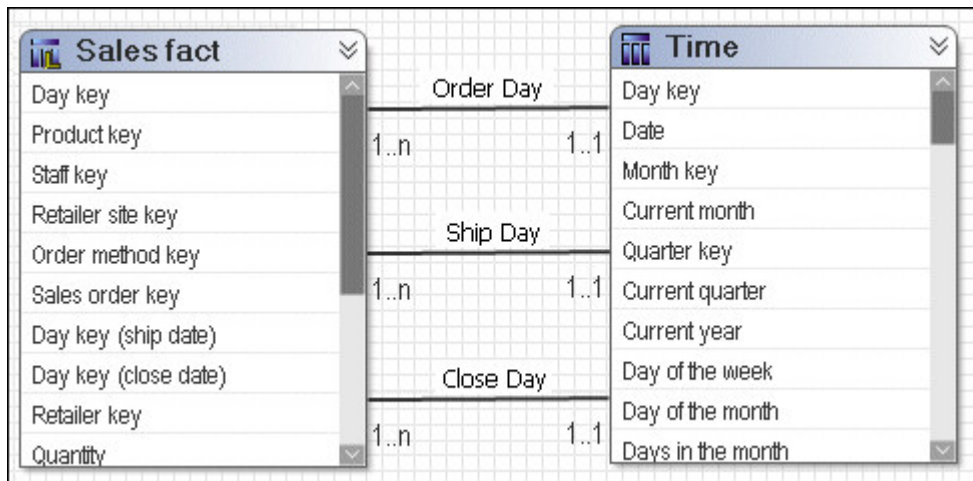
- «Dimensions de rôles»
- «Jointures de boucles», à la page 353
- «Relations réflexives et récursives», à la page 354

Vous pouvez utiliser **Model Advisor** pour détecter les relations qui pourraient poser des problèmes de génération de requêtes et les résoudre en procédant de l'une des façons décrites ci-après. Notez qu'il existe d'autres façons de résoudre les problèmes abordés ici. L'objectif principal est de simplifier les chemins d'accès de requête.

### Dimensions de rôles :

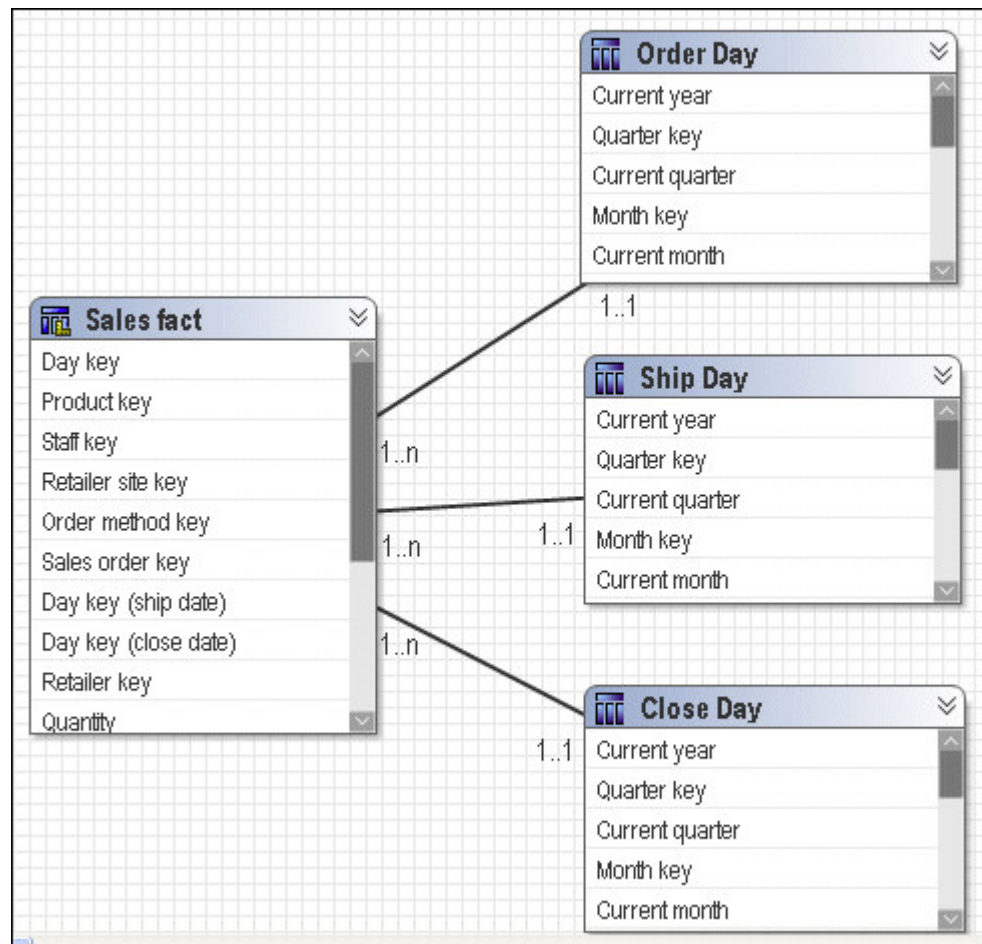
Une table présentant plusieurs relations valides entre elle et une autre table est appelée dimension de rôles. Cela se produit la plupart du temps avec des dimensions telles que Temps et Client.

Par exemple, il existe plusieurs relations entre le fait Ventes et le sujet de requête Temps par le biais des clés Jour de commande, Jour d'expédition et Jour de clôture de la commande.



Supprimez les relations des objets, sujets de requête de faits et sujets de requête de dimensions de rôle importés. Créez un sujet de requête de modèle pour chaque rôle. Eventuellement, pensez à exclure les éléments de requête superflus pour réduire la longueur de l'arborescence de métadonnées affichée pour les utilisateurs. Assurez-vous qu'il existe une relation appropriée unique entre chaque sujet de requête de modèle et le sujet de requête de fait. **Remarque :** Cela va remplacer le paramètre **SQL réduit** mais, compte tenu de la représentation à une seule table de la dimension Temps, ce n'est pas un problème dans le cas présent.

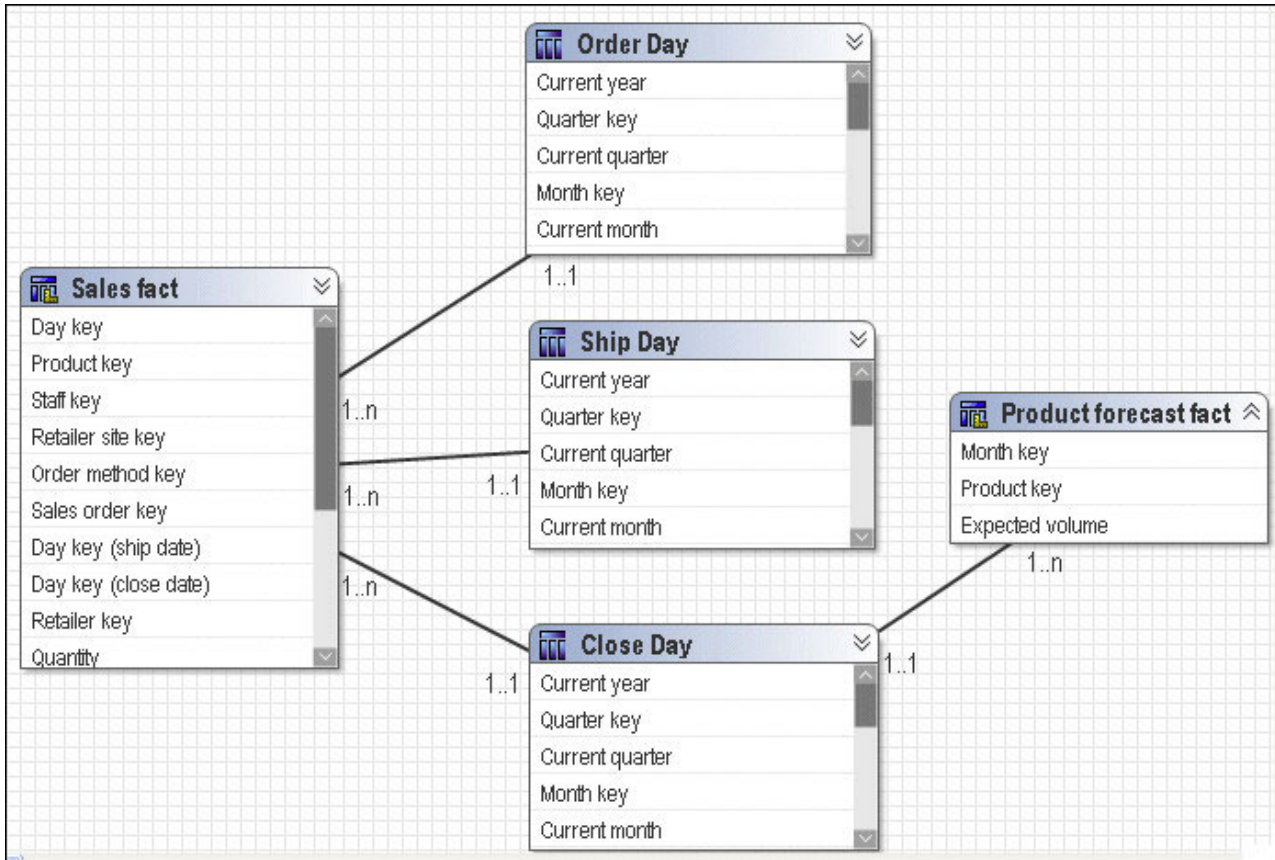




Indiquez de quelle façon utiliser ces rôles avec d'autres faits ne partageant pas les mêmes concepts. Par exemple, le fait Prévisions de vente n'est associé qu'à une seule clé Temps. Vous devez connaître vos données et votre activité pour déterminer si l'ensemble ou une partie des rôles créés pour la dimension Temps sont applicables au fait Prévisions de vente.

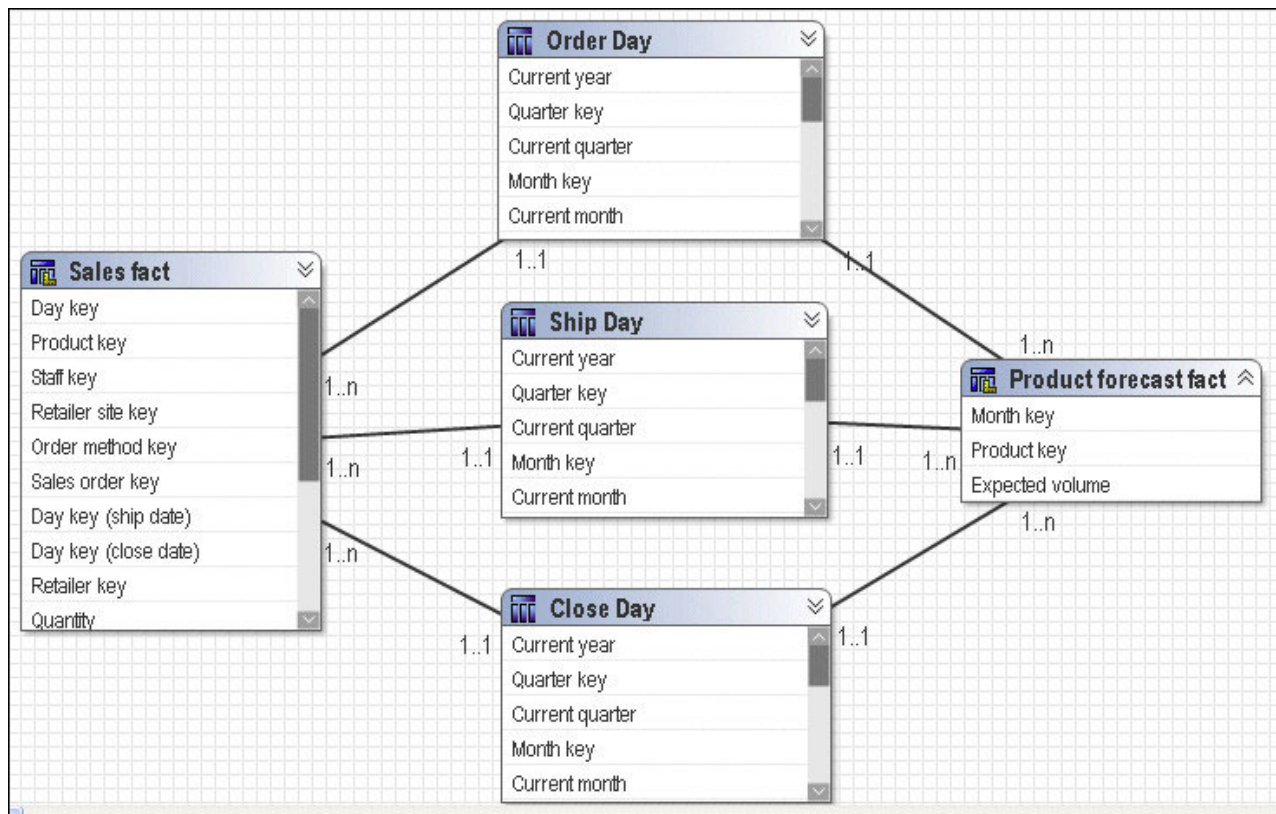
Dans cet exemple, vous pouvez effectuer l'une des opérations suivantes :

- Créer un sujet de requête supplémentaire en tant que dimension de temps conforme et le nommer de façon explicite en tant que dimension conforme. Choisir le rôle que vous allez utiliser le plus. Vous pouvez ensuite vérifier qu'une jointure est établie entre cette version et tous les faits la requérant. Dans cet exemple, c'est Jour de la clôture de la commande qui a été choisi.



- Vous pouvez traiter Jour d'expédition, Jour de la commande et Jour de clôture de la commande comme des sujets de requête de temps interchangeables avec le fait Prévisions de vente.

Dans ce cas, vous devez créer des jointures entre chaque dimension de rôles et le fait Prévisions de vente. Lorsque vous émettez une requête concernant le fait Prévisions de vente, vous ne pouvez utiliser qu'une seule dimension de temps à la fois, sinon votre rapport risque de ne contenir aucune donnée. Par exemple, Month\_key=Clé de mois d'expédition (200401) et Month key=Clé de mois de clôture (200312).



Si vous procédez à une modélisation dimensionnelle, utilisez chaque sujet de requête de modèle en tant que source de dimension ordinaire, puis nommez la dimension et les hiérarchies de façon appropriée. Vérifiez qu'il existe une relation de portée correspondante spécifique pour chaque rôle.

### Jointures de boucles :

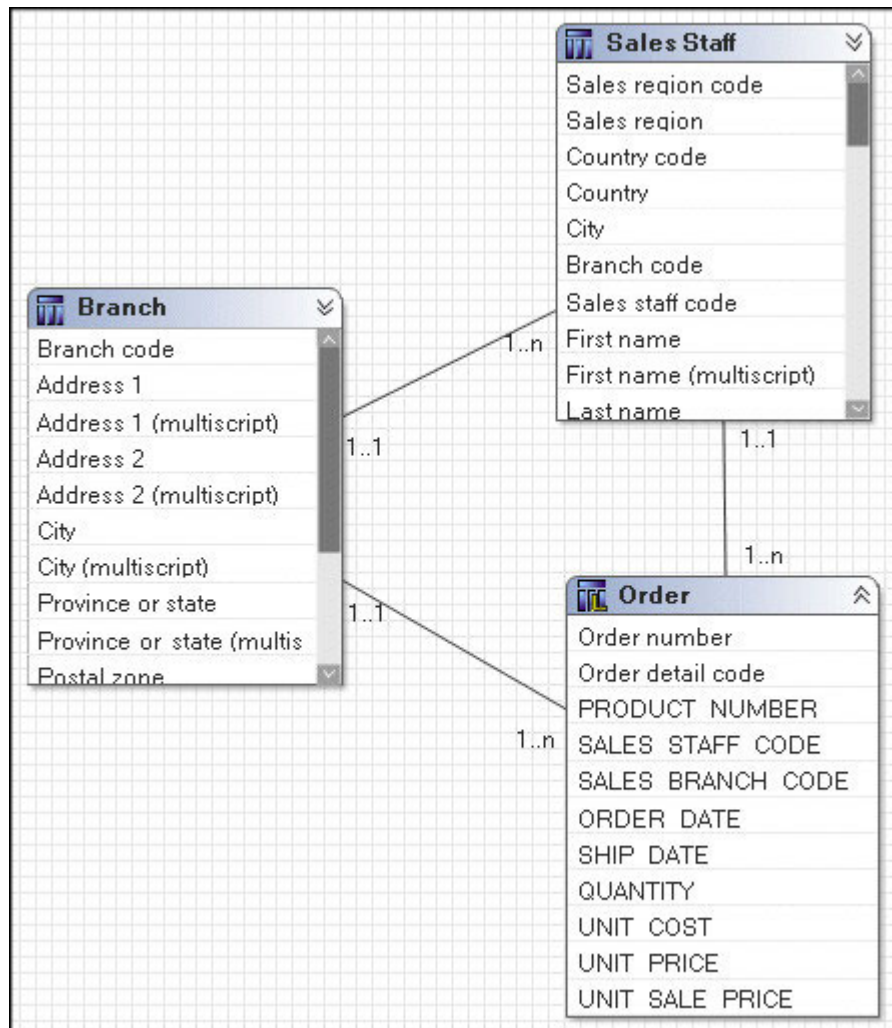
Les jointures de boucles du modèle sont généralement source de comportement imprévisible. Ce n'est pas le cas des jointures de boucles de schémas en étoile.

**Remarque :** Si la cardinalité identifie clairement les faits et dimensions, le logiciel IBM Cognos peut automatiquement résoudre les jointures de boucles causées par des données de schémas en étoile lorsque plusieurs tables de faits sont jointes à un ensemble commun de tables de dimensions.

En cas de jointures de boucles, les premières manifestations du problème sont des sujets de requête définis de façon ambiguë. Lorsque les sujets de requête sont définis de façon ambiguë et font partie d'une jointure de boucle, les jointures utilisées dans une requête donnée sont choisies selon différents facteurs, tels que l'emplacement des relations, le nombre de segments dans les chemins de jointure et, si aucune différence n'est trouvée dans ces éléments, le premier chemin de jointure par ordre alphabétique. Cela est source de confusion chez vos utilisateurs et nous vous conseillons d'identifier clairement les chemins de jointure dans votre modèle.

Sales Staff et Branch offrent un exemple significatif de jointure de boucle avec sujets de requête définis de façon ambiguë.

Dans cet exemple, il est possible de joindre Succursale soit directement à Commande soit par l'intermédiaire de Personnel de vente. Le principal problème est que le résultat obtenu n'est pas le même entre le chemin de jointure Succursale/Commande et le chemin de jointure Succursale/Personnel de vente/Commande. Effectivement, les employés peuvent changer de succursale pendant l'année. Ceux qui l'ont fait font l'objet d'un cumul dans leur nouvelle succursale, même si une grande partie de leurs ventes avaient été effectuée dans leur succursale précédente. Compte tenu de la conception du modèle, il n'y a aucun moyen de savoir lequel des deux chemins de jointure sera utilisé, cela peut varier en fonction des éléments choisis dans la requête.



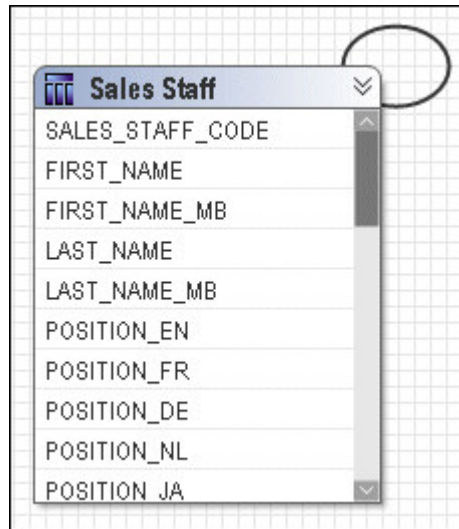
### Relations réflexives et récursives :

Les relations réflexives et récursives impliquent un minimum de deux niveaux de granularité. IBM Cognos Framework Manager importe les relations réflexives mais ne les utilise pas lors de l'exécution de requêtes. Les relations réflexives, qui correspondent à des jointures automatiques, sont affichées dans le modèle à titre représentatif uniquement.

Pour créer une relation réflexive opérationnelle, vous pouvez créer un raccourci d'alias, une copie du sujet de requête de la source de données ou un sujet de requête de modèle. Vous devez ensuite créer une relation entre le sujet de requête d'origine et le nouveau. L'utilisation d'un sujet de requête de modèle s'avère plus

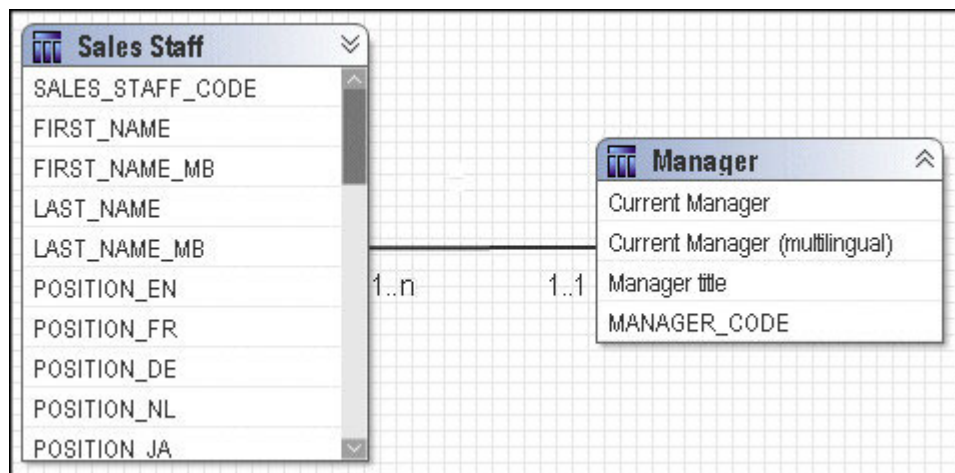
adaptée en termes de souplesse, puisque vous pouvez préciser quels éléments de requête doivent être inclus dans le sujet de requête. Les raccourcis constituent une meilleure solution du point de vue de la maintenance. Pour plus d'informations, voir «Objets de modèle et raccourcis», à la page 342.

Par exemple, le sujet de requête Personnel de vente présente une relation récursive entre les éléments Cd personnel de vente et Code du gestionnaire.



Créez un sujet de requête de modèle pour représenter la catégorie Gestionnaire. Créez une relation de type 1..1 à 1..n entre les catégories Gestionnaire et Personnel de vente. Ensuite, fusionnez-les dans un nouveau sujet de requête de modèle.

Pour une structure simple à deux niveaux utilisant un sujet de requête de modèle Gestionnaire basé sur la catégorie Personnel de vente, le modèle se présente comme suit :



Dans le cas d'une hiérarchie récursive et équilibrée, répétez cette opération pour chaque niveau supplémentaire de la hiérarchie.

Dans le cas d'une hiérarchie non équilibrée ou récursive possédant un grand nombre de niveaux, il est conseillé d'aplanir la hiérarchie dans la source de données et de modéliser la hiérarchie ainsi aplanie en une seule dimension ordinaire.

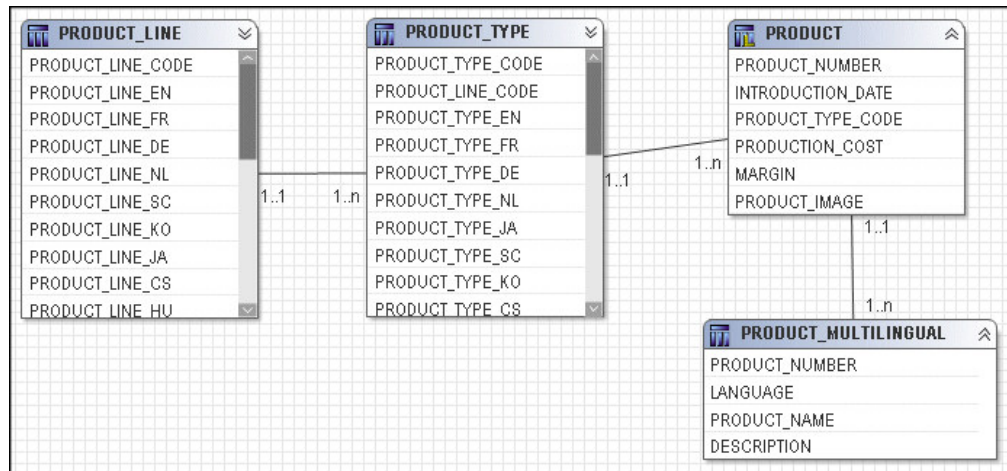
### Simplification du modèle relationnel

Vous pouvez simplifier le modèle en appliquant des concepts de schéma en étoile aux données dimensionnelles et aux données de fait.

#### Modélisation des sujets de requête représentant des données descriptives :

La modélisation dimensionnelle du logiciel IBM Cognos nécessite l'application des principes du schéma en étoile aux couches logiques du modèle.

Les sources de données normalisées ou en flocons sont souvent dotées de plusieurs tables décrivant un concept professionnel unique. Par exemple, une représentation normalisée du concept Produit peut inclure quatre tables liées par des relations de type 1..n. Chaque catégorie Ligne de produits est constituée d'une ou de plusieurs catégories Types de produit. Chaque catégorie Type de produit est constituée d'une ou de plusieurs catégories Produits. Les produits sont associés à des noms et des descriptions dans plusieurs langues, ils existent donc dans la table de recherche Produit multilingue.



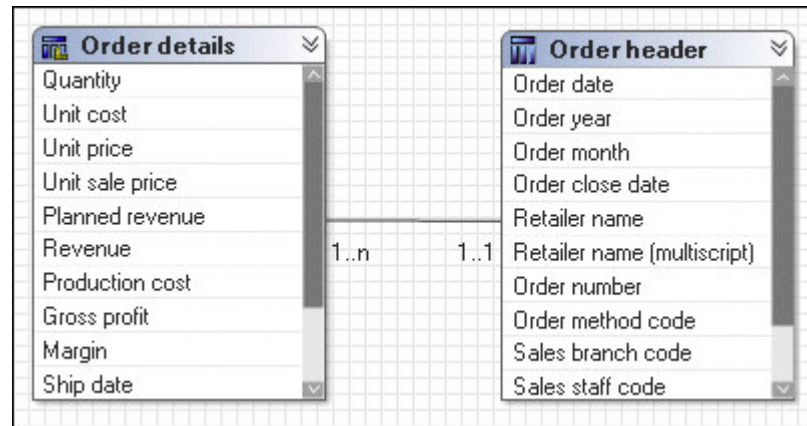
L'une des méthodes de simplification du modèle consiste à créer un sujet de requête de modèle pour chaque concept métier descriptif. Il est possible que vos utilisateurs ne connaissent pas la relation entre les sujets de requête individuels, il est donc utile de les regrouper ; en outre, devoir développer chaque objet de modèle et sélectionner un élément de requête demande plus d'efforts.

L'étape suivante de l'analyse consiste à créer une dimension ordinaire avec un niveau pour chaque sujet de requête.

#### Modélisation de données de fait :

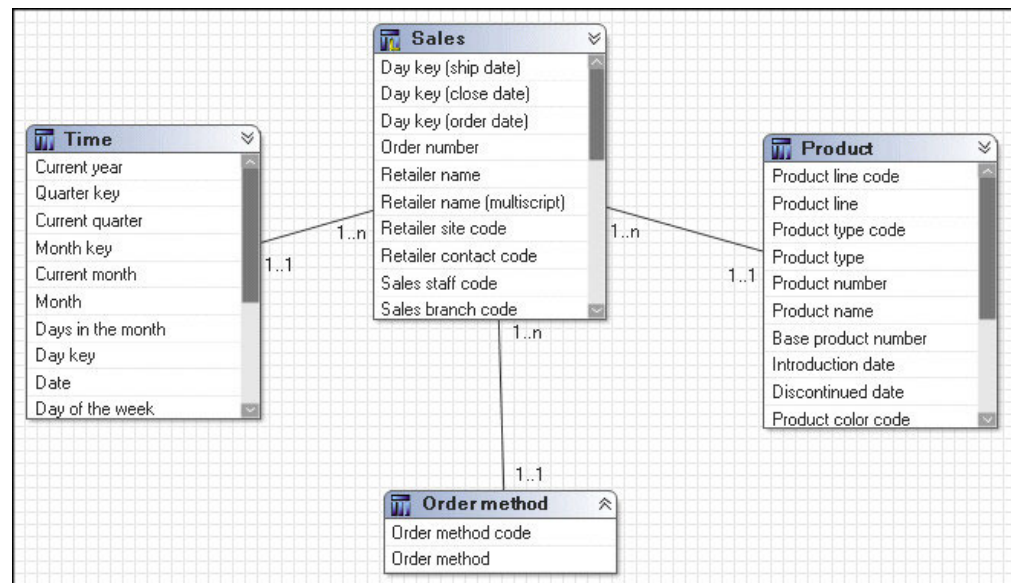
Les sources de données comportent souvent des tables de type Principale/Détails, lesquelles contiennent elles-mêmes des faits. Par exemple, lorsque les tables Order header et Order details sont utilisées pour insérer et mettre à jour des données, la structure Principale/Détails présente des avantages. Lorsque ces tables sont utilisées pour la génération de rapports et l'analyse de données, vous pouvez les regrouper en un seul concept métier logique de façon à simplifier le modèle. Vous

pouvez également insérer une dimension entre elles, Articles retournés, par exemple. Choisissez la solution qui correspond le mieux à vos besoins.



Pour simplifier le modèle de cet exemple, appliquez des concepts de schéma en étoile afin de créer un sujet de requête de modèle combinant les clés étrangères des tables En-têtes de commande et Détails de commande et incluant toutes les mesures au niveau Détails de commande. Ce sujet de requête peut être joint aux mêmes sujets de requête que ceux auxquels les tables En-têtes de commande et Détails de commande ont été joints. Vous pouvez supprimer les relations d'origine des deux sujets de requête de source de données, à l'exception de la relation qui définit la jointure entre eux. Pour connaître les avantages et les inconvénients de la création de relations vis-à-vis de sujets de requête, consultez les exemples fournis dans la section «Définition du langage SQL réduit», à la page 339.

Dans l'exemple ci-dessous, En-têtes de commande et Détails de commande ont été combinés dans un nouveau sujet de requête de modèle nommé Ventes. Ce sujet de requête a été joint à Produit, Temps et Modes de commande.



L'étape suivante de l'analyse consiste à créer une dimension de mesure basée sur le sujet de requête de modèle.

## Définition de la représentation dimensionnelle du modèle

La modélisation dimensionnelle de sources de données relationnelles est possible dans IBM Cognos Framework Manager. Vous pouvez modéliser des dimensions avec des hiérarchies et des niveaux et avoir des faits dotés de plusieurs mesures. Vous pouvez ensuite relier les dimensions aux mesures en définissant la portée du modèle.

Vous devez modéliser de façon dimensionnelle une source de données relationnelle si vous voulez l'utiliser dans IBM Cognos Analysis Studio, activer les passages aux niveaux supérieurs et inférieurs dans les rapports, ou accéder aux fonctions membres dans les studios.

Vous pouvez utiliser le modèle relationnel comme couche de base, puis définir la représentation dimensionnelle du modèle.

Vous pouvez ensuite organiser le modèle pour la présentation. Voir «Organisation du modèle», à la page 361.

### Création de dimensions ordinaires

Une dimension ordinaire contient des informations de clé métier et des informations descriptives classées selon une hiérarchie, du niveau de granularité le plus élevé au plus faible. Elle dispose généralement de plusieurs niveaux qui doivent tous avoir une clé et une légende. Si vous n'avez pas de clé unique pour votre niveau, il est conseillé d'en créer une dans un calcul.

Les dimensions ordinaires de modèle sont basées sur des sources de données ou des sujets de requête de modèle déjà définis dans le modèle. Vous devez définir une clé métier et une légende de type de chaîne pour chaque niveau. Lorsque vous vérifiez le modèle, toute absence de clé métier et de légende est détectée. Au lieu de joindre des dimensions ordinaires de modèle aux dimensions de mesure, créez des jointures sur les sujets de requête sous-jacents et créez une relation de portée entre la dimension ordinaire et la dimension de mesure.

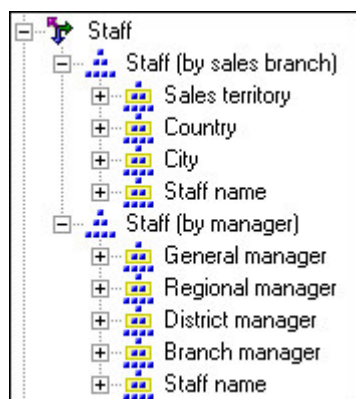
### Modélisation de dimensions avec plusieurs hiérarchies

Des hiérarchies multiples sont créées lorsque différentes vues structurées peuvent s'appliquer aux mêmes données. Selon la nature des hiérarchies et des rapports requis, vous aurez peut-être besoin de vérifier la technique de modélisation appliquée, selon le cas.

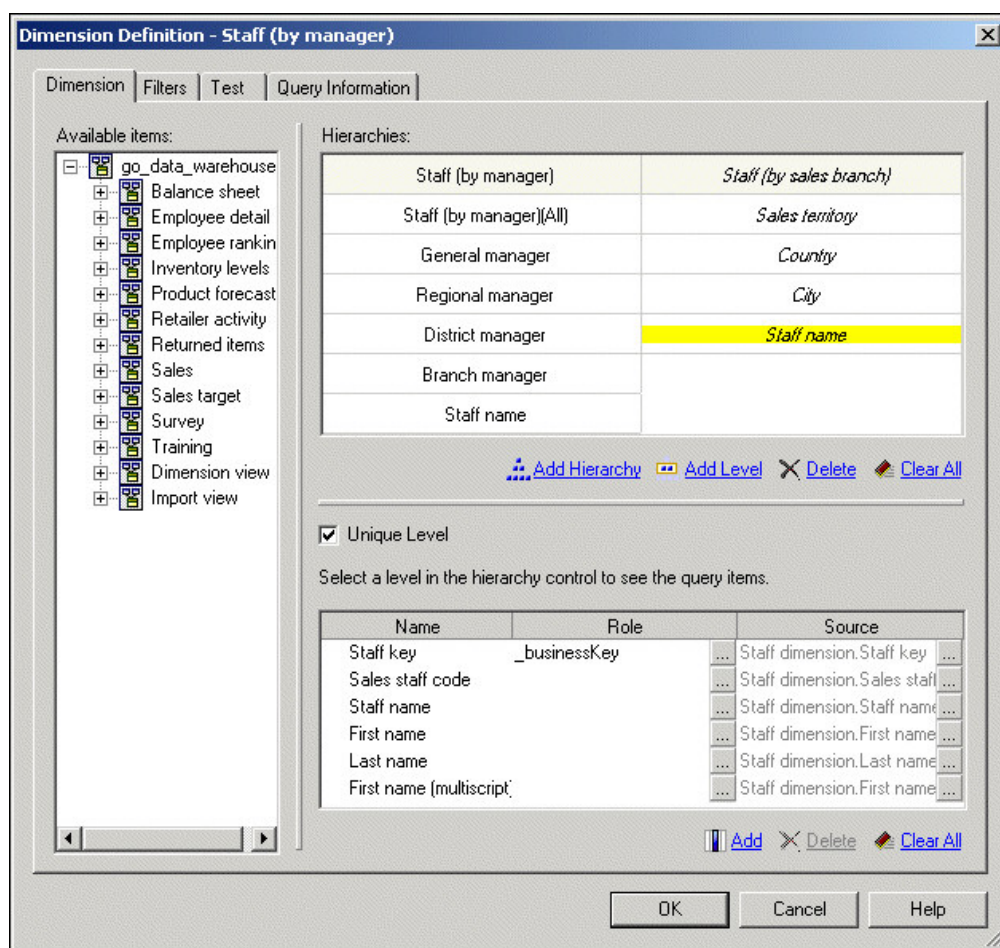
Par exemple, l'équipe commerciale peut être visualisée par responsable ou par zone géographique. Dans les studios d'IBM Cognos, ces hiérarchies sont des structures logiques séparées, mais interchangeable, liées à la même requête sous-jacente.

Voici une dimension unique d'équipe commerciale avec deux hiérarchies :





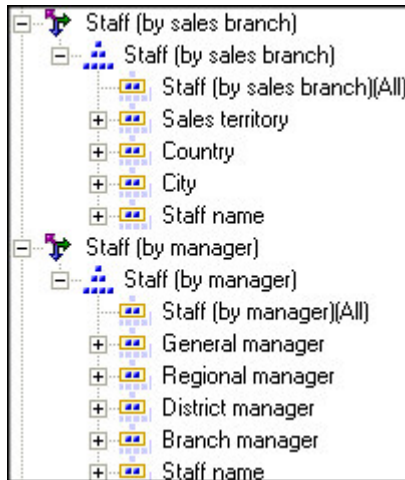
Les hiérarchies sont définies comme suit dans Framework Manager :



Dans Framework Manager, vous pouvez indiquer plusieurs hiérarchies pour des dimensions ordinaires. Plusieurs hiérarchies définies pour une dimension ordinaire se comportent comme des vues d'une même requête. Néanmoins, vous ne pouvez utiliser qu'une seule hiérarchie à la fois dans une requête. Par exemple, vous ne pouvez pas utiliser une hiérarchie sur les lignes d'un rapport de type tableau croisé et une autre hiérarchie de la même dimension dans les colonnes. Si vous avez besoin de ces deux hiérarchies dans le même rapport, vous devez créer deux dimensions, une pour chaque hiérarchie. Si vous disposez de plusieurs hiérarchies avec des niveaux ou agrégations sensiblement différents, vous pouvez effectuer

voire modélisation de sorte qu'un sujet de requête distinct, avec les déterminants appropriés, serve de base à cette hiérarchie. La seule obligation est de définir une jointure entre le sujet de requête utilisé en tant que base de la hiérarchie et le sujet de requête qui fournit les données de faits.

Voici des dimensions distinctes pour chaque hiérarchie :



Utilisez cette approche si des ensembles de colonnes sensiblement différents correspondent à chaque hiérarchie et s'il s'avère plus intuitif pour vos utilisateurs de modéliser les hiérarchies en tant que dimensions distinctes, avec des requêtes distinctes et plus simples.

## Création de dimensions de mesure

Une dimension de mesure est une série de faits. Vous pouvez créer une dimension de mesure pour un ou plusieurs sujets de requête entre lesquels il existe une relation valide.

Les dimensions de mesure de modèle doivent être exclusivement composées d'éléments quantitatifs. Par nature, il n'existe pas de clés sur lesquelles appliquer la jointure, il est donc impossible de créer des jointures vers des dimensions de mesure de modèle. Au lieu de joindre des dimensions de mesure de modèle aux dimensions ordinaires, créez des jointures au niveau des sujets de requête sous-jacents. Ensuite, créez manuellement une relation de portée entre eux ou bien détectez cette portée si les deux dimensions se trouvent dans le même espace-noms.

## Création de relations de portée

Les relations de portée sont établies uniquement entre des dimensions de mesure et des dimensions ordinaires pour définir le niveau auquel les mesures sont disponibles à des fins de génération de rapports. Elles sont différentes des jointures et n'ont aucune incidence sur la clause Where. Dans une relation de portée, aucune condition ni critère ne sont définis sur la façon de former une requête. Elle indique uniquement si une dimension peut faire l'objet de requêtes avec un fait spécifique. En l'absence de relation de portée, une erreur se produit lors de l'exécution.

Si vous définissez la relation de portée pour la dimension de mesure, les mêmes paramètres s'appliquent à toutes les mesures de la dimension de mesure. Si les données font l'objet de rapports à un niveau différent des mesures dans la

dimension de mesure, vous pouvez définir la portée d'une mesure. Vous pouvez définir le niveau le plus bas à partir duquel les données peuvent faire l'objet d'un rapport.

Lorsque vous créez une dimension de mesure, IBM Cognos Framework Manager crée une relation de portée entre la dimension de mesure et chaque dimension ordinaire existante. Framework Manager recherche les chemins de jointure entre la dimension de mesure et les dimensions ordinaires, en commençant par le niveau de détail le plus bas. Si de nombreux chemins de jointure sont disponibles, la relation de portée créée par Framework Manager peut ne pas être celle prévue. Dans ce cas, vous devez éditer la relation de portée.

## Organisation du modèle

Après avoir utilisé la base du modèle relationnel et créé une représentation dimensionnelle, vous pouvez organiser le modèle.

- Conservez les métadonnées de la source de données dans un espace-noms ou dossier distinct.
- Créez au moins un espace-noms ou dossier facultatif pour résoudre les difficultés relatives à l'interrogation à l'aide de sujets de requête.  
Pour utiliser IBM Cognos Analysis Studio, il doit y avoir un espace-noms ou dossier dans le modèle qui représente les métadonnées avec objets dimensionnels.
- Créez au moins un espace-noms ou dossier pour la vue métier globale des métadonnées contenant des raccourcis vers les dimensions ou sujets de requête.  
Utilisez des concepts métier pour modéliser la vue métier. Un même modèle peut contenir de nombreuses vues métier, chacune adaptée à un groupe d'utilisateurs différent. Publiez les vues métier.
- Créez «Groupes sous forme de schémas en étoile».
- Appliquez la sécurité des objets, si nécessaire.
- Créez des packs et publiez les métadonnées.

Pour plus d'informations sur les sujets non abordés ici, voir le document Framework Manager *User Guide*.

### Groupes sous forme de schémas en étoile

Le concept des dimensions conformes n'est pas limité à la modélisation dimensionnelle, il s'applique également aux sujets de requête.

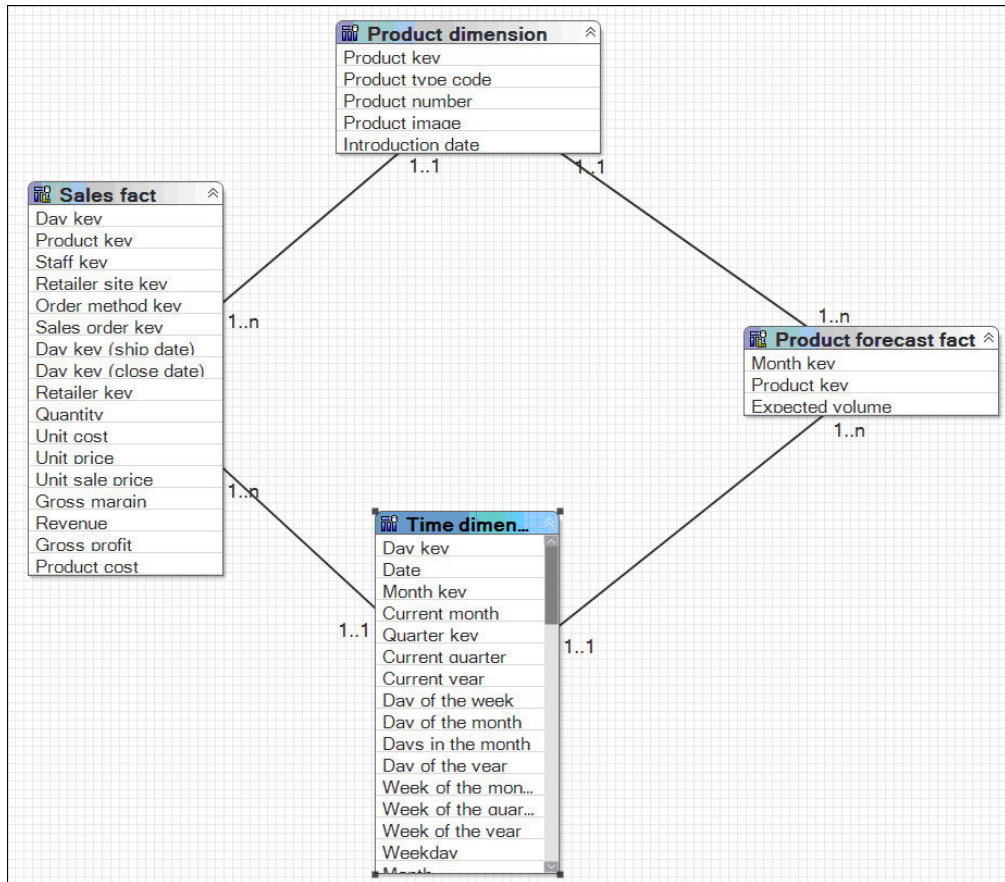
Utilisez l'assistant **Créer un regroupement de schémas en étoile** pour créer rapidement des groupes de raccourcis qui vont fournir du contexte à vos utilisateurs sur les appartenances des objets entre eux. Cela rend le modèle plus intuitif aux yeux des utilisateurs. Les groupes sous forme de schémas en étoile peuvent également faciliter la génération de rapports portant sur des faits multiples en permettant la répétition des dimensions partagées dans différents groupes. Cela aide les utilisateurs à voir ce que les différents groupes ont en commun et la façon dont ils peuvent générer des rapports inter-fonctionnels ou à faits multiples. Pour plus d'informations, voir «Requêtes à granularité multiple et à faits multiples», à la page 335.

Les groupes sous forme de schémas en étoile fournissent également du contexte pour les requêtes à plusieurs chemins de jointure. En créant des groupes sous forme de schémas en étoile dans la vue métier du modèle, vous pouvez déterminer quel chemin de jointure doit être sélectionné lorsqu'il y en a beaucoup. Cette fonctionnalité est particulièrement utile dans le cadre de requêtes sans fait.

## Schémas en étoiles conformes multiples ou requêtes sans faits :

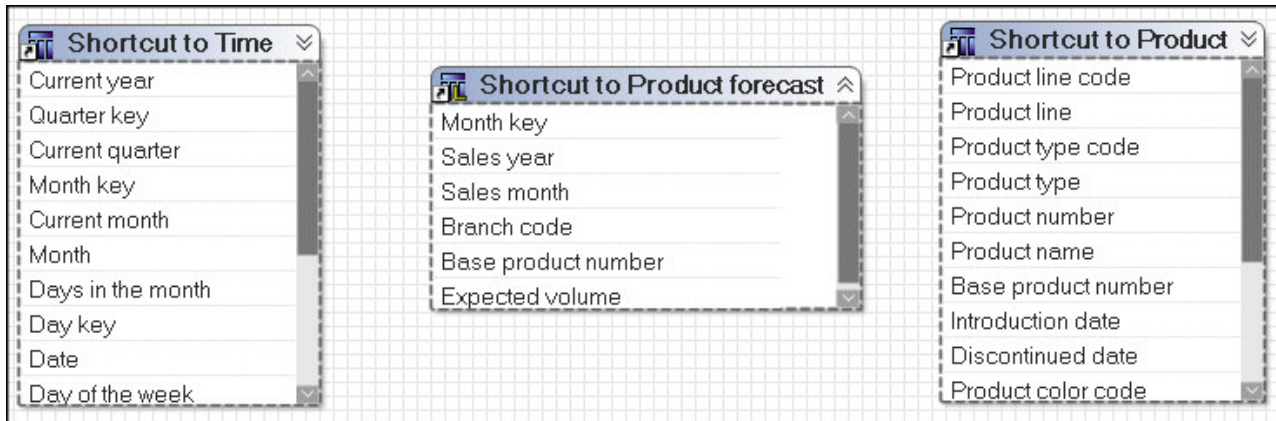
Vous verrez probablement des sujets de requête dimensionnels qui sont joints à plusieurs sujets de requête de faits. L'ambiguïté au niveau des jointures est un problème pouvant se produire lors de la création de rapports à partir d'éléments issus de plusieurs dimensions ou sujets de requête dimensionnels sans inclure aucun élément de la dimension de mesure ou du sujet de requête de fait. On parle alors de requête sans fait.

Par exemple, les dimensions Produit et Temps sont associées aux faits Prévisions de vente et Ventas.



A l'aide de ces relations, comment créer un rapport utilisant uniquement des éléments de Produit et Année ? La question professionnelle pourrait être de savoir quels produits ont été prévus à la vente en 2005 ou quels produits ont effectivement été vendus en 2005. Même si cette requête implique uniquement Produit et Temps, ces dimensions sont associées par le biais de plusieurs faits. Il n'existe aucun moyen de savoir quelle question professionnelle est posée. Vous devez définir le contexte pour la requête sans fait.

Dans l'exemple suivant, il est conseillé de créer deux espaces-noms, un contenant des raccourcis vers Produit, Temps et Prévisions de vente, l'autre contenant des raccourcis vers Produit, Temps et Ventas.



Lorsque vous procédez ainsi pour tous les schémas en étoile, vous résolvez les ambiguïtés de jointure en plaçant les raccourcis vers le fait et toutes les dimensions dans un même espace-noms. Les raccourcis associés aux dimensions conformes de chaque espace-noms sont identiques et constituent des références vers l'objet d'origine. **Remarque** : Il s'agit exactement de la même règle que celle appliquée aux dimensions ordinaires et aux dimensions de mesure.

Avec un espace-noms pour chaque schéma en étoile, vos utilisateurs n'ont plus aucun doute concernant les éléments à utiliser. Pour créer un rapport sur les produits réellement vendus en 2005, ils utilisent les éléments Produit et Année de l'espace-noms Ventés. La seule relation pertinente dans ce contexte est celle associant les éléments Produit, Temps et Ventés ; elle est utilisée pour renvoyer les données.



---

## Chapitre 10. Code SQL généré par IBM Cognos Software

Le code SQL généré par IBM Cognos Software est souvent difficile à interpréter. Le présent document offre une explication du code SQL obtenu lors de situations courantes.

**Remarque :** Les exemples de code SQL indiqués dans le présent document servent à illustrer des cas de figure spécifiques ; leur longueur a été éditée. Ces exemples utilisent la version 8.2 de l'exemple de modèle.

Pour accéder au *Guide de modélisation des métadonnées IBM Cognos* dans une autre langue, accédez à `emplacement_installation\c10\webcontent\documentation` et ouvrez le dossier correspondant à la langue choisie. Ouvrez ensuite le fichier `ug_best.pdf`.

---

### Présentation des requêtes dimensionnelles

Les requêtes dimensionnelles sont conçues pour permettre l'utilisation de requêtes à faits multiples.

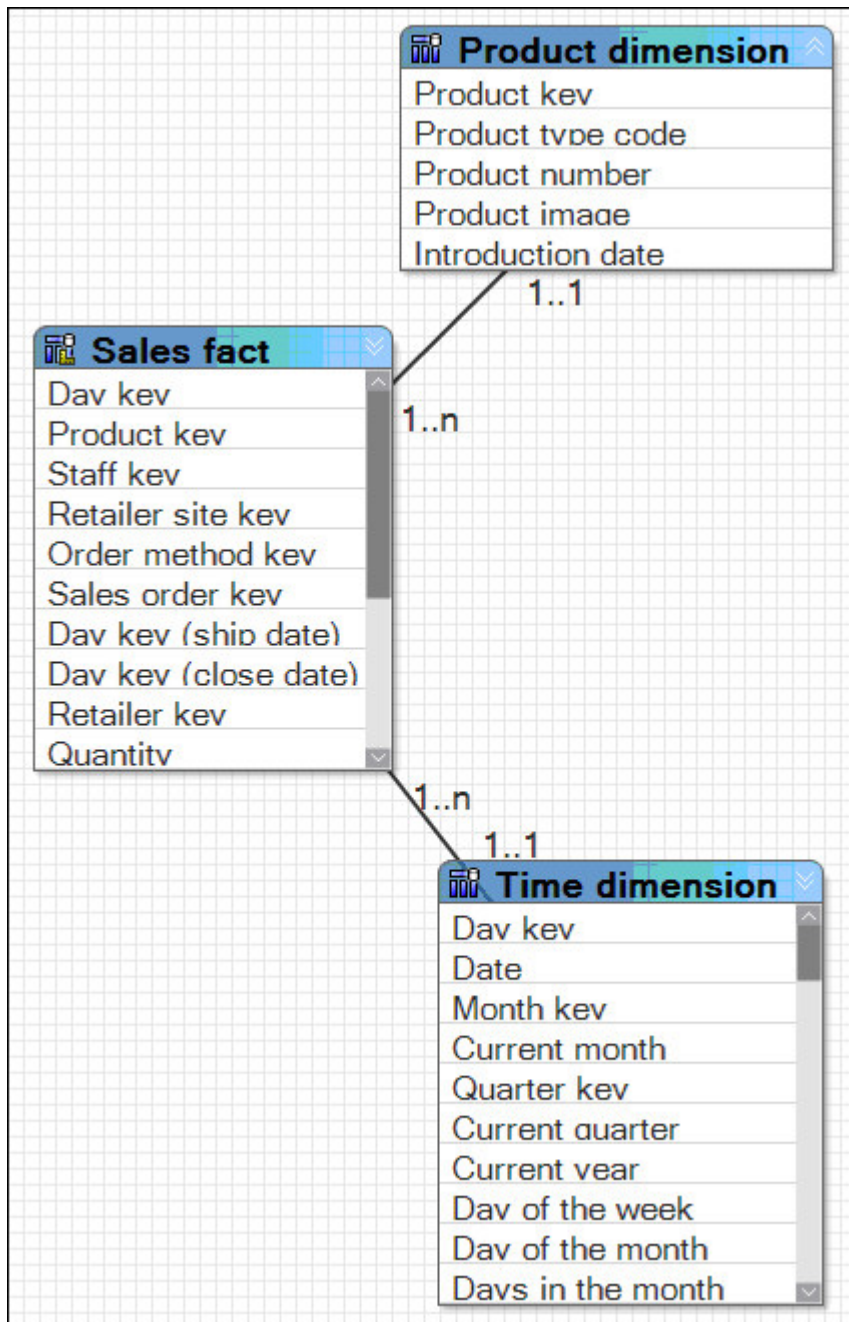
Les objectifs de base des requêtes à faits multiples sont les suivants :

- Conserver les données lorsque les données de faits ne sont pas parfaitement alignées entre deux dimensions communes, par exemple lorsqu'il y a plus de lignes dans les faits que dans les dimensions.
- Eviter le double comptage lorsque des données de faits existent à différents niveaux de granularité, en s'assurant que chaque fait est représenté dans une seule requête avec le type de groupement approprié. Dans certains cas, vous devrez peut-être créer des déterminants pour les sujets de requête sous-jacents.

#### Requête à fait unique

Lorsqu'une requête figure dans un groupe sous forme de schéma en étoile, le résultat obtenu est une requête à fait unique.

Dans cet exemple, toutes les requêtes utilisent le fait Sales (Ventes). Les dimensions fournissent des attributs et des descriptions qui rendent les données du fait Sales plus explicites. Toutes les relations entre les dimensions et ce fait sont de type 1-n.



Lorsque vous appliquez un filtre sur le mois et le produit, vous obtenez le résultat suivant :

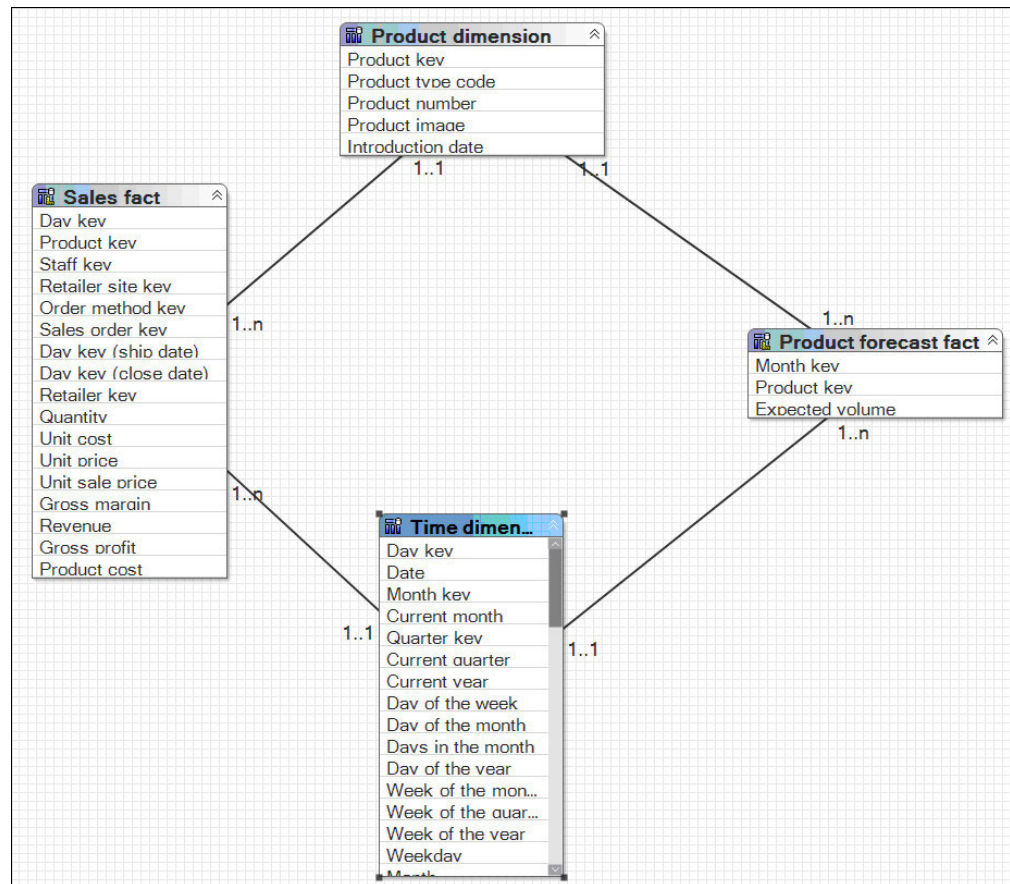
MONTH_NAME	PRODUCT_NAME	QUANTITY
April 2004	Aloe Relief	1,410
April 2004	Course Pro Umbrella	132
February 2004	Aloe Relief	270
February 2006	Aloe Relief	88



## Requête à granularité multiple et à faits multiples sur des dimensions conformes

Une requête portant sur des faits multiples et sur des dimensions conformes respecte la cardinalité entre chaque table de faits et ses dimensions et écrit des éléments de code SQL pour renvoyer toutes les lignes à partir de chaque table de faits.

Par exemple, Ventes et Prévisions relatives au produit sont des faits.



Notez qu'il ne s'agit pas d'un exemple mais d'une représentation simplifiée de la façon dont ces données s'affichent dans un modèle créé à l'aide des recommandations de modélisation d'IBM Cognos.

### Résultat

Des requêtes individuelles portant sur les faits Ventes et Prévisions relatives au produit par mois et par produit génèrent les résultats suivants. Les données du fait Ventes sont stockées au niveau des jours.

MONTH_NAME	PRODUCT_NAME	EXPECTED_VOLUME
April 2004	Aloe Relief	1,690
April 2004	Course Pro Umbrella	125
February 2004	Aloe Relief	246
February 2004	Course Pro Umbrella	1
February 2006	Aloe Relief	92

Une requête portant sur les faits Ventes et Prévisions relatives au produit respecte la cardinalité entre chaque table de faits et ses dimensions et écrit des éléments de code SQL pour renvoyer toutes les lignes à partir de chaque table de faits. Les tables de faits sont mises en correspondance d'après leurs clés, mois et produit communs, et dans la mesure du possible, elles sont agrégées au niveau de granularité commun le plus bas. Dans ce cas, les jours sont cumulés pour constituer des mois. Des valeurs nulles sont souvent renvoyées pour ce type de requête car une combinaison d'éléments dimensionnels figurant dans une table de faits peut ne pas exister dans l'autre table de faits.

MONTH_NAME	PRODUCT_NAME	QUANTITY	EXPECTED_VOLUME
April 2004	Aloe Relief	1,410	1,690
April 2004	Course Pro Umbrella	132	125
February 2004	Aloe Relief	270	246
February 2004	Course Pro Umbrella		1
February 2006	Aloe Relief	88	92

Notez que en février 2004, l'élément Parapluie Omnium faisait partie des prévisions mais aucun article de cette ligne n'a été vendu. Les données des faits Ventes et Prévisions relatives au produit existent à différents niveaux de granularité. Les données du fait Ventes existent au niveau des jours, et celles du fait Prévisions relatives au produit figurent au niveau des mois.

## Code SQL

Le code SQL généré par le logiciel IBM Cognos, appelé requête liée, est souvent difficile à interpréter. Une requête liée utilise plusieurs sous-requêtes, une pour chaque étoile, rassemblées par une jointure externe complète sur les clés communes. L'objectif est de conserver tous les membres dimensionnels qui se trouvent de chaque côté de la requête.

L'exemple suivant a été modifié en termes de longueur ; il est utilisé comme exemple de capture des fonctions principales des requêtes liées.

```
select
  coalesce(D2.MONTH_NAME,D3.MONTH_NAME) as MONTH_NAME,
  coalesce(D2.PRODUCT_NAME,D3.PRODUCT_NAME) as PRODUCT_NAME,
  D2.EXPECTED_VOLUME as EXPECTED_VOLUME,
  D3.QUANTITY as QUANTITY
from (select TIME.MONTH_NAME as MONTH_NAME,
  PRODUCT_LOOKUP.PRODUCT_NAME as PRODUCT_NAME,
  XSUM(PRODUCT_FORECAST_FACT.EXPECTED_VOLUME for
  TIME.CURRENT_YEAR,TIME.QUARTER_KEY,TIME.MONTH_KEY,
  PRODUCT.PRODUCT_LINE_CODE, PRODUCT.PRODUCT_TYPE_CODE,
  PRODUCT.PRODUCT_KEY) as EXPECTED_VOLUME
```

```

from
  (select TIME.CURRENT_YEAR as CURRENT_YEAR,
    TIME.QUARTER_KEY as QUARTER_KEY,
    TIME.MONTH_KEY as MONTH_KEY,
    XMIN(TIME.MONTH_NAME for TIME.CURRENT_YEAR,
    TIME.QUARTER_KEY,TIME.MONTH_KEY) as MONTH_NAME
  from TIME_DIMENSION TIME
  group by TIME.MONTH_KEY) TIME
join PRODUCT_FORECAST_FACT PRODUCT_FORECAST_FACT
on (TIME.MONTH_KEY = PRODUCT_FORECAST_FACT.MONTH_KEY)
join PRODUCT_PRODUCT on (PRODUCT.PRODUCT_KEY =
PRODUCT_FORECAST_FACT.PRODUCT_KEY)
where
  (PRODUCT.PRODUCT_NAME in ('Aloe Relief','Course Pro
Umbrella')) and
  (TIME.MONTH_NAME in ('April 2004','February 2004','February
2006'))
group by
  TIME.MONTH_NAME,
  PRODUCT_LOOKUP.PRODUCT_NAME
) D2
full outer join
(select TIME.MONTH_NAME as MONTH_NAME,
  PRODUCT_LOOKUP.PRODUCT_NAME as PRODUCT_NAME,
  XSUM(SALES_FACT.QUANTITY for TIME.CURRENT_YEAR,
  TIME.QUARTER_KEY, TIME.MONTH_KEY,
  PRODUCT.PRODUCT_LINE_CODE, PRODUCT.PRODUCT_TYPE_CODE,
  PRODUCT.PRODUCT_KEY ) as QUANTITY
from
  select TIME.DAY_KEY,TIME.MONTH_KEY,TIME.QUARTER_KEY,
  TIME.CURRENT_YEAR,TIME.MONTH_EN as MONTH_NAME
  from TIME_DIMENSION TIME) TIME
join SALES_FACT SALES_FACT
on (TIME.DAY_KEY = SALES_FACT.ORDER_DAY_KEY)
join PRODUCT_PRODUCT on (PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY)
where
  PRODUCT.PRODUCT_NAME in ('Aloe Relief','Course Pro Umbrella'))
  and (TIME.MONTH_NAME in ('April 2004','February 2004','February
2006'))
group by
  TIME.MONTH_NAME,
  PRODUCT.PRODUCT_NAME
) D3
on ((D2.MONTH_NAME = D3.MONTH_NAME) and
(D2.PRODUCT_NAME = D3.PRODUCT_NAME))

```

## Qu'est-ce qu'une instruction Coalesce ?

Une instruction coalesce est simplement un procédé efficace permettant de traiter les éléments de requêtes à partir de dimensions conformes. Elle est utilisée pour accepter la première valeur non nulle retournée par l'un des sujets de requête. Cette instruction accepte une liste exhaustive de clés sans répétition lors de l'application d'une jointure externe complète.

## A quoi sert une jointure externe complète ?

Une jointure externe complète est nécessaire pour garantir que toutes les données de chaque table de faits sont extraites. Une jointure interne ne donne des résultats que si un article en stock a été vendu. Une jointure externe sur l'élément de droite renvoie toutes les ventes relatives aux articles en stock. Une jointure externe sur l'élément de gauche renvoie tous les articles en stock ayant fait l'objet d'une vente. Une jointure externe complète est le seul moyen permettant de savoir ce qui était en stock et ce qui a été vendu.

## Modélisation de relations de type 1-n en relations de type 1-1

Si une relation de type 1-n existe dans les données, mais qu'elle est modélisée en tant que relation de type 1-1, les interruptions SQL sont inévitables, car les informations fournies au logiciel IBM Cognos par les métadonnées ne sont pas suffisantes.

Les problèmes qui se produisent le plus couramment lorsque des relations 1-n sont modélisées en tant que relations 1-1 sont les suivants :

- Le double comptage des requêtes à granularité multiple n'est pas empêché automatiquement.

IBM Cognos ne parvient pas à détecter les faits et à générer une requête liée pour compenser le double comptage, ce qui peut se produire avec des relations hiérarchiques et différents niveaux de granularité entre les dimensions conformes.

- Les requêtes à faits multiples ne sont pas détectées automatiquement.

IBM Cognos ne dispose pas de suffisamment d'informations pour détecter une requête à faits multiples. Pour les requêtes à faits multiples, une jointure interne est effectuée et la jointure de boucle est éliminée en supprimant la dernière jointure évaluée. La suppression d'une jointure est susceptible de générer des résultats incorrects ou imprévisibles selon les dimensions et faits inclus dans la requête.

Si la cardinalité est modifiée de façon à utiliser uniquement des relations 1-1 entre les sujets de requête ou les dimensions, le résultat d'une requête portant sur les faits Prévisions relatives au produit et Ventes avec les faits Temps ou Temps et Produits génère une seule instruction Select visant à supprimer une jointure pour éviter une référence circulaire.

L'exemple ci-dessous montre que les résultats de cette requête sont incorrects si on les compare aux résultats des requêtes individuelles portant sur les faits Ventes ou Prévisions relatives au produit.

Les résultats des requêtes individuelles sont répertoriés ci-après.

MONTH_NAME	PRODUCT_NAME	QUANTITY
April 2004	Aloe Relief	1,410
April 2004	Course Pro Umbrella	132
February 2004	Aloe Relief	270
February 2006	Aloe Relief	88

MONTH_NAME	PRODUCT_NAME	EXPECTED_VOLUME
April 2004	Aloe Relief	1,690
April 2004	Course Pro Umbrella	125
February 2004	Aloe Relief	246
February 2004	Course Pro Umbrella	1
February 2006	Aloe Relief	92

Lorsque vous associez ces requêtes en une seule, vous obtenez les résultats ci-après.

MONTH_NAME	PRODUCT_NAME	QUANTITY	EXPECTED_VOLUME
April 2004	Aloe Relief	68,598	1,811,680
April 2004	Course Pro Umbrella	68,598	134,000
February 2004	Aloe Relief	29,672	105,780
February 2004	Course Pro Umbrella	29,672	430
February 2006	Aloe Relief	28,564	47,196

## Code SQL

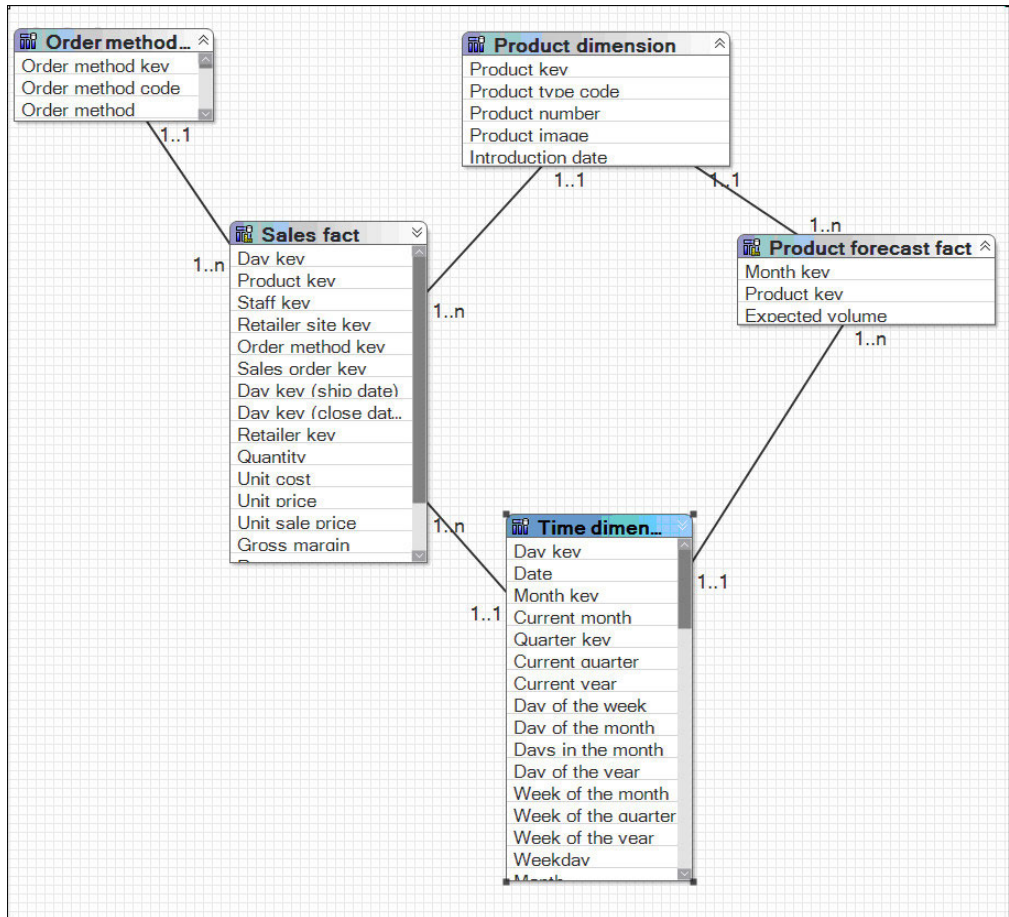
En raison d'un chemin de jointure circulaire détecté dans le modèle, le code SQL généré n'a pas inclus l'une des relations non requise pour compléter le chemin de jointure. Dans cet exemple, la relation entre Temps et Prévisions relatives au produit a été supprimée.

Un chemin de jointure circulaire génère rarement une requête qui produit des résultats utiles.

```
select
  TIME.MONTH_NAME as MONTH_NAME,
  PRODUCT_LOOKUP.PRODUCT_NAME as PRODUCT_NAME,
  XSUM(SALES_FACT.QUANTITY for
  TIME.CURRENT_YEAR, TIME.QUARTER_KEY, TIME.MONTH_KEY,
  PRODUCT.PRODUCT_LINE_CODE, PRODUCT.PRODUCT_TYPE_CODE,
  PRODUCT.PRODUCT_KEY ) as QUANTITY,
  XSUM(PRODUCT_FORECAST_FACT.EXPECTED_VOLUME for TIME.CURRENT_YEAR,
  TIME.QUARTER_KEY, TIME.MONTH_KEY, PRODUCT.PRODUCT_LINE_CODE,
  PRODUCT.PRODUCT_TYPE_CODE, PRODUCT.PRODUCT_KEY ) as EXPECTED_VOLUME
from
  (select TIME.DAY_KEY,TIME.MONTH_KEY, TIME.QUARTER_KEY,
  TIME.CURRENT_YEAR,TIME.MONTH_EN as MONTH_NAME
  from TIME_DIMENSION TIME) TIME
join
  SALES_FACT on (TIME.DAY_KEY = SALES_FACT.ORDER_DAY_KEY)
join
  PRODUCT_FORECAST_FACT on (TIME.MONTH_KEY =
  PRODUCT_FORECAST_FACT.MONTH_KEY)
join
  PRODUCT (PRODUCT.PRODUCT_KEY = PRODUCT_FORECAST_FACT.PRODUCT_KEY)
where
  (PRODUCT.PRODUCT_NAME in ('Aloe Relief','Course Pro Umbrella'))
and
  (TIME.MONTH_NAME in ('April 2004','February 2004','February 2006'))
group by
  TIME.MONTH_NAME, PRODUCT.PRODUCT_NAME
```

## Requête à granularité multiple et à faits multiples sur des dimensions non conformes

Si une dimension non conforme est ajoutée à la requête, la nature du résultat renvoyé par la requête liée change. Il n'est plus possible de regrouper les enregistrements à un niveau de granularité commun plus bas, car un côté de la requête présente un caractère dimensionnel qui n'est pas commun avec l'autre côté de la requête. Le résultat renvoyé est formé de deux listes en corrélation.



## Résultat

Les résultats de requêtes individuelles portant sur les schémas en étoile respectifs se présentent comme suit :

MONTH_NAME	PRODUCT_NAME	ORDER_METHOD	QUANTITY
April 2004	Aloe Relief	E-mail	114
April 2004	Aloe Relief	Fax	220
April 2004	Aloe Relief	Mail	100
April 2004	Aloe Relief	Sales visit	322
April 2004	Aloe Relief	Telephone	286
April 2004	Aloe Relief	Web	368
April 2004	Course Pro Umbrella	E-mail	22
April 2004	Course Pro Umbrella	Fax	28
April 2004	Course Pro Umbrella	Sales visit	82
February 2004	Aloe Relief	Mail	28
February 2004	Aloe Relief	Sales visit	102
February 2004	Aloe Relief	Telephone	28
February 2004	Aloe Relief	Web	112
February 2006	Aloe Relief	Sales visit	88

MONTH_NAME	PRODUCT_NAME	QUANTITY	EXPECTED_VOLUME
April 2004	Aloe Relief	1,410	1,690
April 2004	Course Pro Umbrella	132	125
February 2004	Aloe Relief	270	246
February 2004	Course Pro Umbrella		1
February 2006	Aloe Relief	88	92

Une requête portant sur les mêmes éléments des deux schémas en étoile produit le résultat suivant.

MONTH_NAME	PRODUCT_NAME	ORDER_METHOD	QUANTITY	EXPECTED_VOLUME
April 2004	Aloe Relief	Sales visit	322	1,690
April 2004	Aloe Relief	Telephone	286	1,690
April 2004	Aloe Relief	Web	368	1,690
April 2004	Aloe Relief	E-mail	114	1,690
April 2004	Aloe Relief	Fax	220	1,690
April 2004	Aloe Relief	Mail	100	1,690
April 2004	Course Pro Umbrella	E-mail	22	125
April 2004	Course Pro Umbrella	Fax	28	125
April 2004	Course Pro Umbrella	Sales visit	82	125
February 2004	Aloe Relief	Web	112	246
February 2004	Aloe Relief	Mail	28	246
February 2004	Aloe Relief	Sales visit	102	246
February 2004	Aloe Relief	Telephone	28	246
February 2004	Course Pro Umbrella			1
February 2006	Aloe Relief	Sales visit	88	92

Dans ce résultat, le niveau de granularité inférieur pour les enregistrements du fait Ventes renvoie un nombre plus important d'enregistrements pour chaque combinaison de mois et de produit. Il existe désormais une relation 1-n entre les lignes renvoyées par le fait Prévisions relatives au produit et celles renvoyées par le fait Ventes.

Lorsque vous comparez cette situation au résultat obtenu dans l'exemple de la requête granularité multiple et à faits multiples sur des dimensions conformes, vous constatez que davantage d'enregistrements sont renvoyés et que les résultats relatifs au fait Volume prévu sont répétés dans plusieurs dimensions Mode de commande. Le fait d'ajouter la dimension Mode de commande à la requête modifie effectivement la relation entre la colonne Quantité et la colonne Volume prévu et la remplace par une relation 1-n. Il n'est plus possible de relier une valeur unique de la colonne Volume prévu à une valeur de la colonne Quantité.

Le fait d'appliquer un groupement à la clé Mois démontre que le résultat obtenu dans cet exemple repose sur le même jeu de données que celui du résultat de la requête à granularité multiple et à faits multiples, mais avec un degré de granularité plus élevé.

## Code SQL

Le code SQL lié généré pour cet exemple est très semblable au code SQL créé dans la requête à faits multiples et à granularité multiple. La principale différence réside dans l'ajout du mode de commande. La dimension Mode de commande n'est pas conforme et affecte uniquement la requête portant sur la table de faits Ventes.

```
select
  D2.QUANTITY as QUANTITY,
  D3.EXPECTED_VOLUME as EXPECTED_VOLUME,
  coalesce(D2.PRODUCT_NAME,D3.PRODUCT_NAME) as PRODUCT_NAME,
  coalesce(D2.MONTH_NAME,D3.MONTH_NAME) as MONTH_NAME,
  D2.ORDER_METHOD as ORDER_METHOD
from
  (select
    PRODUCT.PRODUCT_NAME as PRODUCT_NAME,
    TIME.MONTH_NAME as MONTH_NAME,
```



```

ORDER_METHOD.ORDER_METHOD as ORDER_METHOD,
XSUM(SALES_FACT.QUANTITY for TIME.CURRENT_YEAR,TIME.QUARTER_KEY,
TIME.MONTH_KEY,PRODUCT.PRODUCT_LINE_CODE,PRODUCT.PRODUCT_TYPE_CODE,
PRODUCT.PRODUCT_KEY,ORDER_METHOD_DIMENSION.ORDER_METHOD_KEY) as
QUANTITY
from
PRODUCT_DIMENSION PRODUCT
join
SALES_FACT SALES_FACT
on (PRODUCT.PRODUCT_KEY = SALES_FACT.PRODUCT_KEY)
join
ORDER_METHOD_DIMENSION ORDER_METHOD
on (ORDER_METHOD.ORDER_METHOD_KEY = SALES_FACT.ORDER_METHOD_KEY)
join TIME_DIMENSION TIME
on ( TIME.DAY_KEY = SALES_FACT.ORDER_DAY_KEY)
where
(PRODUCT.PRODUCT_NAME in ('Aloe Relief','Course Pro Umbrella'))
and
( TIME.MONTH_NAME in ('April 2004','February 2004','February 2006'))
group by
PRODUCT.PRODUCT_NAME,
TIME.MONTH_NAME,
ORDER_METHOD.ORDER_METHOD
) D2
full outer join
(select
PRODUCT.PRODUCT_NAME as PRODUCT_NAME,
TIME.MONTH_NAME as MONTH_NAME,
XSUM(PRODUCT_FORECAST_FACT.EXPECTED_VOLUME for TIME.CURRENT_YEAR,
TIME.QUARTER_KEY,TIME.MONTH_KEY,PRODUCT.PRODUCT_LINE_CODE,
PRODUCT.PRODUCT_TYPE_CODE,PRODUCT.PRODUCT_KEY) as EXPECTED_VOLUME
from
PRODUCT_DIMENSION PRODUCT
join
PRODUCT_FORECAST_FACT PRODUCT_FORECAST_FACT
on (PRODUCT.PRODUCT_KEY = PRODUCT_FORECAST_FACT.PRODUCT_KEY)
join
(select
TIME.CURRENT_YEAR as CURRENT_YEAR,
TIME.QUARTER_KEY as QUARTER_KEY,
TIME.MONTH_KEY as MONTH_KEY,
XMIN(TIME.MONTH_NAME for TIME.CURRENT_YEAR, TIME.QUARTER_KEY,
TIME.MONTH_KEY) as MONTH_NAME
from
TIME_DIMENSION TIME
group by
TIME.CURRENT_YEAR,
TIME.QUARTER_KEY,
TIME.MONTH_KEY
) TIME
on (TIME.MONTH_KEY = PRODUCT_FORECAST_FACT.MONTH_KEY)
where
(PRODUCT.PRODUCT_NAME in ('Aloe Relief','Course Pro Umbrella'))
and
(TIME.MONTH_NAME in ('April 2004','February 2004','February 2006'))
group by
PRODUCT.PRODUCT_NAME,
TIME.MONTH_NAME
) D3
on ((D2.PRODUCT_NAME = D3.PRODUCT_NAME) and
(D2.MONTH_NAME = D3.MONTH_NAME))

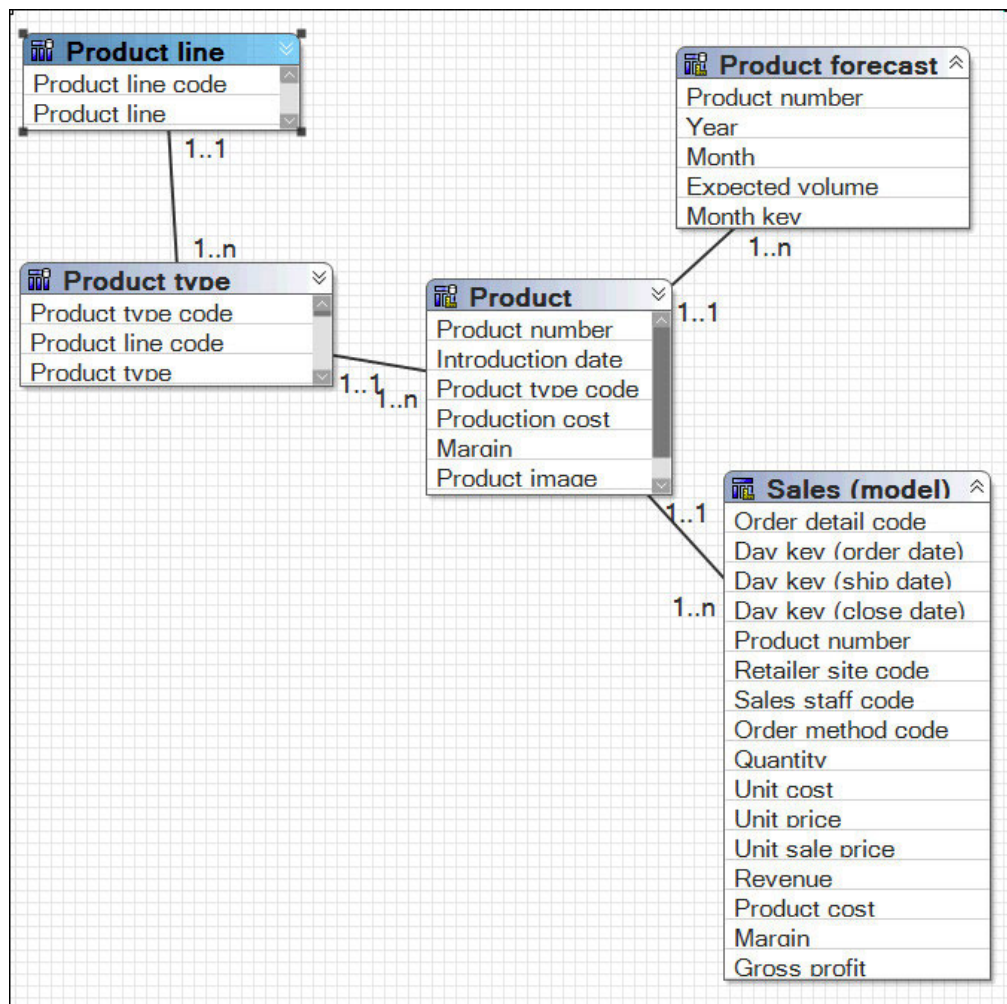
```

## Résolutions des dimensions et faits identifiés de façon ambiguë

Un sujet de requête est considéré comme étant défini de façon ambiguë s'il participe à la fois à la relation n et à la relation 1 avec d'autres sujets de requête. Un sujet de requête défini de façon ambiguë n'est pas nécessairement dangereux du point de vue de la création de requêtes. Nous vous conseillons d'utiliser les cas présentés ci-dessous pour évaluer les sujets de requête. Le but de cette évaluation est d'éviter tout fractionnement de requête non nécessaire et de vérifier que les fractionnements qui se produisent sont intentionnels et corrects.

### Sujets de requête représentant un niveau de hiérarchie

Un cas fréquent de sujet de requête défini de façon ambiguë qui n'est pas dangereux est celui où le sujet de requête représente un niveau intermédiaire d'une hiérarchie descriptive. Un exemple en est la Hiérarchie Produits qui suit.



Dans cet exemple, les dimensions Type de produit et Produit pourraient être identifiées comme étant définies de façon ambiguë, mais cette ambiguïté n'est négative ni pour les résultats produits, ni pour les performances des requêtes faisant appel à un ou plusieurs de ces sujets de requête. Vous n'avez pas besoin de corriger ce modèle de requête car d'après les règles de détection de fait, un seul fait est identifié dans n'importe quelle requête qui associe un élément des sujets de

requête Prévisions de vente ou Ventes. Il est néanmoins recommandé de réduire les hiérarchies en une seule dimension ordinaire lors d'une modélisation à des fins d'analyse.

Exemples de requêtes pouvant être écrites à l'aide de cet exemple :

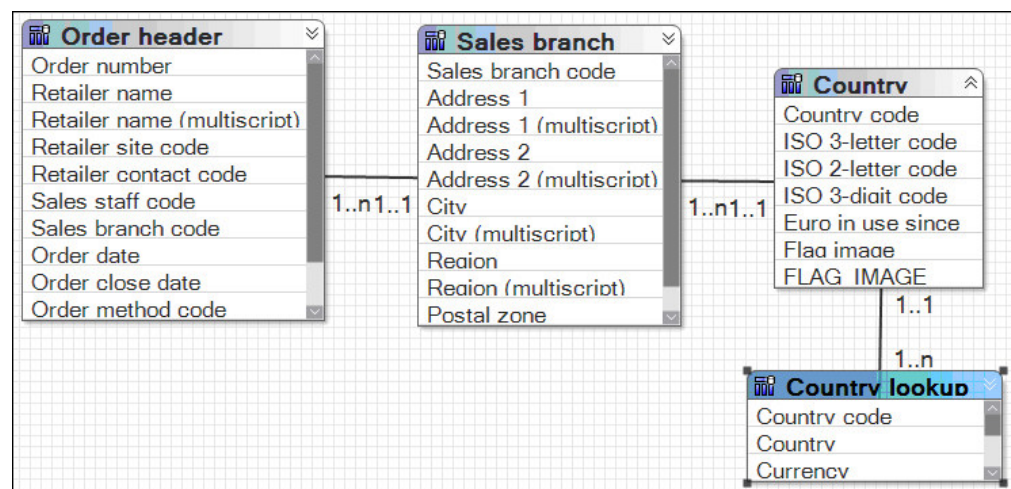
Les éléments des sujets de requête ci-dessous sont utilisés dans une requête :	Sujet de requête se comportant comme un fait dans la requête :
Lignes de produits et Type de produit	Type de produit
Lignes de produits, Type de produit et Produit	Produit
Lignes de produits, Type de produit, Produit et Ventes	Ventes
Lignes de produits et Ventes	Ventes
Lignes de produits et Prévisions de vente	Prévisions de vente

## Résolution de requêtes fractionnées par erreur

Si des requêtes sont fractionnées alors qu'elles n'auraient pas dû l'être, vous devez les résoudre.

Les sujets de requêtes situés du côté n de toutes les relations sont identifiés comme étant des faits. Dans l'exemple qui suit, vous constatez que les sujets Order Header et Country Multilingual se comportent comme des faits. En réalité, le sujet de requête Country Multilingual ne contient que des informations descriptives et sert vraisemblablement de table de recherche. Selon une perspective de dimension ou de modélisation métier, le sujet de requête Country Multilingual est une extension du sujet Country.

En quoi est-ce un problème de laisser le modèle tel quel ?



Testez ce modèle en créant un rapport relatif au nombre de commandes par ville puis par pays ou région. Lorsque vous utilisez ce modèle, vous obtenez un résultat incorrect. Les chiffres sont exacts pour les villes mais certaines villes ne sont pas associées au bon pays ou à la région correcte. Voici un exemple de résultat lié de façon incorrecte.

COUNTRY	CITY	Number of Orders
Australia	Amsterdam	279
Austria	Bilbao	123
Belgium	Birmingham	164
Brazil	Boston	515
Canada	Calgary	123

Habituellement, il s'agit du point de départ lorsque vous voyez ce genre de résultat dans le code SQL.

## Code SQL

Dans cet exemple, vous voyez une requête liée, ce qui est logique si le modèle contient des faits multiples. Par définition, une requête liée est une requête qui tente de lier plusieurs faits entre eux. Elle utilise les relations qui existent entre les faits, ainsi que les déterminants des dimensions conformes (ou communes) définies dans le modèle. Une requête liée peut être identifiée par deux requêtes disposant d'une jointure externe intégrale. Une requête d'encapsuleur doit inclure une instruction coalesce portant sur les dimensions conformes.

Notez les problèmes suivants dans le code SQL :

- La requête ne dispose d'aucune instruction coalesce.
- RSUM indique une tentative de création d'une clé valide.

```

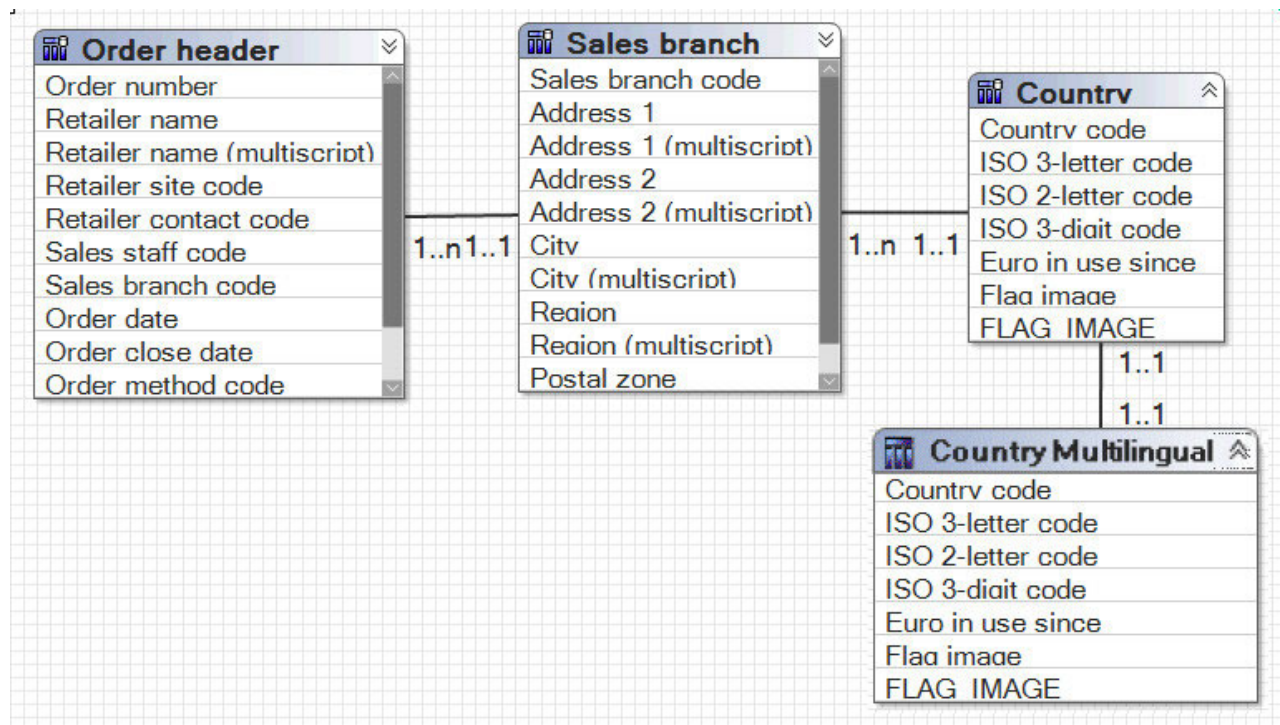
select
  D3.COUNTRY as COUNTRY,
  D2.CITY as CITY,
  D2.number_of_orders as number_of_orders
from
  (select
    SALES_BRANCH.CITY as CITY,
    XCOUNT(ORDER_HEADER.ORDER_NUMBER for SALES_BRANCH.CITY) as
    number_of_orders,
    RSUM(1 at SALES_BRANCH.CITY order by SALES_BRANCH.CITY
    asc local)
    as sc
  from
    gosales.gosales.dbo.SALES_BRANCH SALES_BRANCH
  join
    gosales.gosales.dbo.ORDER_HEADER ORDER_HEADER
  on (SALES_BRANCH.SALES_BRANCH_CODE = ORDER_HEADER.SALES_BRANCH_CODE)
  group by
    SALES_BRANCH.CITY
  order by
    CITY asc
  ) D2
full outer join
  (select
    COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
    RSUM(1 at COUNTRY_MULTILINGUAL.COUNTRY order by
    COUNTRY_MULTILINGUAL.COUNTRY asc local) as sc
  from
    gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
  group by
    COUNTRY_MULTILINGUAL.COUNTRY
  order by
    COUNTRY asc
  ) D3
on (D2.sc = D3.sc)

```

En examinant les colonnes liées de chaque requête, vous constatez qu'elles sont calculées selon des critères non connexes. Cela explique pourquoi il n'existe apparemment aucune relation entre les pays ou les régions et les villes cités dans le rapport.

Pourquoi une requête liée s'affiche-t-elle ? Pour répondre à cette question, vous devez examiner le modèle.

Dans cet exemple, les éléments de requête utilisés dans le rapport provenaient de différents sujets de requête. Country ou region provenait de Country Multilingual, City de Sales Branch et Number of Orders d'un nombre issu de Order Number dans le sujet de requête Order Header.



Le problème vient du fait que la requête est fractionnée parce que le moteur de requêtes la considère comme une requête à faits multiples. Toutefois, le fractionnement n'est pas associé à une clé valide à partir de laquelle établir une liaison, puisque aucun élément n'est commun aux deux faits.

Pour résoudre ce problème, vous disposez de deux méthodes qui nécessitent toutes les deux une bonne compréhension des données.

### Solution 1

Vous pouvez ajouter un filtre au sujet Country Multilingual pour changer la cardinalité de la relation en 1-1.

```
Select *
from [GOSL].COUNTRY_MULTILINGUAL
Where
COUNTRY_MULTILINGUAL."LANGUAGE"='EN'
```

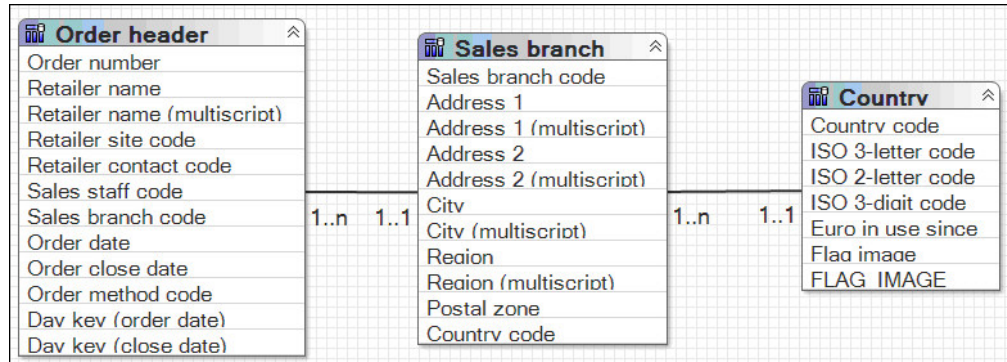
Vous pouvez également ajouter un filtre portant sur la relation et remplacer la cardinalité par 1-1.

COUNTRY.COUNTRY\_CODE = COUNTRY\_MULTILINGUAL.COUNTRY\_CODE  
 and COUNTRY\_MULTILINGUAL.LANGUAGE = 'EN'

Ces deux possibilités entraînent la création d'un modèle possédant un fait unique pour cette requête.

## Solution 2

Simplifiez le modèle en regroupant les sujets de requête connexes. Cette solution est très avantageuse, puisque le modèle est simplifié et les possibilités d'erreur liées à la création de requêtes sont réduites.



Que vous optiez pour l'une ou l'autre de ces solutions, le résultat de la requête est maintenant correct.

COUNTRY	CITY	Number of Orders
Australia	Melbourne	98
Austria	Wien	162
Belgium	Heverlee	94
Brazil	São Paulo	15
Canada	Calgary	123
Canada	Toronto	330

Le code SQL ne correspond plus à une requête liée.

```
select
  Country.c7 as COUNTRY,
  SALES_BRANCH.CITY as CITY,
  XCOUNT(ORDER_HEADER.ORDER_NUMBER for Country.c7,SALES_BRANCH.CITY)
  as number_of_orders
from
  (select
    COUNTRY.COUNTRY_CODE as c1,
    COUNTRY_MULTILINGUAL.COUNTRY as c7
  from
    gosales.gosales.dbo.COUNTRY COUNTRY
  join
    gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
  on (COUNTRY.COUNTRY_CODE = COUNTRY_MULTILINGUAL.COUNTRY_CODE)
  where COUNTRY_MULTILINGUAL.LANGUAGE='EN'
  ) Country
join
```

```
gosales.gosales.dbo.SALES_BRANCH SALES_BRANCH
on (SALES_BRANCH.COUNTRY_CODE = Country.c1)
join
gosales.gosales.dbo.ORDER_HEADER ORDER_HEADER
on (SALES_BRANCH.SALES_BRANCH_CODE = ORDER_HEADER.SALES_BRANCH_CODE)
group by
Country.c7,
SALES_BRANCH.CITY
```





---

## Chapitre 11. Mise à niveau de modèles

La mise à niveau d'un modèle le prépare pour lui permettre de bénéficier des nouvelles fonctionnalités de la présente édition.

Lorsque vous installez la dernière version d'IBM Cognos Software, la plupart des modèles créés avec des versions précédentes sont mis à niveau automatiquement. Les packs que vous avez publiés pour des utilisateurs sont automatiquement mis à jour dans la base de données Content Manager. Ces packs continueront de fonctionner comme auparavant et tous les rapports continueront de s'exécuter. Mettez le modèle à niveau dans IBM Cognos Framework Manager uniquement si vous souhaitez modifier des métadonnées publiées.

Si la mise à niveau d'un modèle n'aboutit pas, les rapports qui utilisent ce modèle ne s'exécuteront pas. Vous devez alors ouvrir le modèle dans Framework Manager, le mettre à niveau manuellement et le republier.

Le processus de mise à niveau des modèles varie selon que vous effectuez une mise à niveau d'un modèle créé dans IBM Cognos ReportNet ou dans des versions précédentes d'IBM Cognos BI.

Pour mettre à niveau le modèle, procédez comme suit :

- Vérifiez le modèle avant la mise à niveau.
- Ouvrez et mettez le modèle à niveau.
- Vérifiez et réparez le modèle ReportNet mis à niveau.

Pour trouver la documentation produit la plus à jour, y compris toutes les documentations traduites, accédez à la page IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

---

### Vérification des modèles avant la mise à niveau

Le processus de mise à niveau ne résout pas les problèmes qui existaient dans votre modèle avant la mise à niveau. Avant de commencer la mise à niveau, vérifiez votre modèle et corrigez tout problème signalé dans l'environnement existant.

Lors d'une mise à niveau depuis IBM Cognos ReportNet vers IBM Cognos BI, assurez-vous que les informations de dimension sont correctement définies dans le modèle ReportNet. Pour plus d'informations, voir «Conversion d'informations de dimension en déterminants ou en dimensions», à la page 386.

---

### Ouverture et la mise à niveau des modèles

Lorsque vous installez la dernière version d'IBM Cognos Software, la plupart des modèles créés avec des versions précédentes sont mis à niveau automatiquement.

Lors de la mise à niveau du modèle, vous devez :

- Comprendre l'impact des restrictions qui sont définies lors de la mise à niveau. Pour plus d'informations, voir «Mise à niveau et restrictions», à la page 384.
- Vérifier les types de données modifiés.

Pour plus d'informations, voir «Mise à niveau et types de données», à la page 385.

- Comprendre l'impact de la mise à niveau sur les sujets de requête basés sur des métadonnées SAP BW.

Pour plus d'informations, voir «Mise à niveau et sujets de requêtes basés sur des métadonnées SAP BW», à la page 385.

Pour plus d'informations, voir «Mise à niveau de projets segmentés et liés», à la page 391.

## Mise à niveau et restrictions

Plusieurs restrictions sont définies automatiquement pendant le processus de mise à niveau :

- La restriction **Autoriser la portabilité du modèle amélioré lors de l'exécution** («Autoriser la portabilité du modèle amélioré lors de l'exécution», à la page 314) est sélectionnée.

Cette restriction est utilisée lors de la mise à niveau d'un modèle ReportNet 1.x. Elle empêche une application rigide des types de données de sorte qu'un modèle IBM Cognos BI puisse fonctionner comme un modèle ReportNet 1.x jusqu'à ce que vous mettiez à jour les types de données dans les métadonnées. Après avoir vérifié que le modèle a été mis à niveau avec succès, désélectionnez cette restriction.

La propriété de statut de tous les sujets de requête de source de données est définie sur **Réévaluation requise**. Le moteur de requête ne tient pas compte des types de données dans le modèle qui sont marqués avec **Réévaluation requise** et extrait les informations de type de données de la source de données. Cela peut avoir un impact légèrement négatif sur les performances lors de l'exécution de rapports.

- La restriction **Autoriser la génération dynamique des informations de dimension** («Autoriser la génération dynamique des informations de dimension», à la page 315) est sélectionnée.

Elle garantit un comportement cohérent avec ReportNet 1.x en dérivant des sortes d'informations de dimension à partir des relations, des informations de clé et des informations d'index de la source de données.

Lorsque cette restriction est sélectionnée et qu'un modèle contient des sujets de requête, mais pas de dimension, IBM Cognos BI génère des requêtes qui évitent le double comptage. Cela garantit également la compatibilité avec ReportNet. Lorsque vous vérifiez le modèle, un avertissement s'affiche lorsque cette restriction est sélectionnée.

- La restriction **Traitement des raccourcis** («Traitement des raccourcis», à la page 311) a pour valeur **Automatique**.

Cette restriction contrôle la façon dont IBM Cognos BI traite les raccourcis. Lorsque la restriction est définie sur **Automatique**, le modèle fonctionne de la même façon que dans ReportNet, c'est-à-dire qu'un raccourci qui se trouve dans le même répertoire que sa cible se comporte comme un alias ou une instance indépendante. En revanche, un raccourci qui se trouve ailleurs dans le modèle se comporte comme une référence au raccourci d'origine. Lorsque vous créez un modèle, cette restriction a toujours la valeur **Jointure explicite**.

- La restriction **Supprimer les valeurs nulles pour les sources de données SAP BW** («Supprimer les valeurs nulles pour les sources de données SAP BW», à la page 315) est sélectionnée pour les métadonnées SAP BW uniquement.

Cette restriction contrôle la création des jointures externes dans les modèles basés sur des métadonnées SAP BW.

Certaines requêtes peuvent être volumineuses car les valeurs nulles ne sont pas supprimées par filtrage. La suppression des valeurs nulles permet de supprimer une ligne ou une colonne dont toutes les valeurs sont nulles (vides).

SAP BW effectue la suppression des valeurs nulles. Cela réduit la quantité de données transférées vers les studios IBM Cognos et améliore les performances.

## Mise à niveau et types de données

IBM Cognos Business Intelligence prend en charge des types de données qui ne sont pas disponibles dans IBM Cognos ReportNet. Cela peut avoir un impact sur la façon dont les types de données sont mappés lors de l'importation de métadonnées pendant la mise à niveau.

**Remarque :** le mappage des types de données varie en fonction du fournisseur de source de données.

Le tableau ci-après présente les principales différences entre les types de données dans ReportNet et IBM Cognos BI.

ReportNet	IBM Cognos BI
char	nChar
decimal	numeric
varChar	nVarChar
varChar	timestampTZ
varChar	IntervalTZ

Les types de données sont stockés dans le modèle de manière à éviter à IBM Cognos BI d'avoir à accéder à la source de données chaque fois qu'il doit déterminer les types de données des éléments de requête. Cela a pour conséquence d'améliorer les performances.

Certains éléments de requête, la plupart des calculs, peuvent apparaître comme étant rompus après la mise à niveau. Cela peut se produire dans les cas suivants :

- Le type de données des éléments sous-jacents d'un calcul a été modifié et il n'est pas possible d'attribuer automatiquement le type de données au calcul.
- Un calcul enregistré contenant une invite n'a pas été correctement testé.

Vous devez tester les calculs affectés et corriger les anomalies.

## Mise à niveau et sujets de requêtes basés sur des métadonnées SAP BW

Un sujet de requête qui est basé sur des métadonnées SAP BW n'est pas converti en une dimension. En revanche, le processus de mise à niveau crée une dimension masquée et un nouveau sujet de requête de modèle qui portent le même nom et contiennent les mêmes éléments de requête que le sujet de requête d'origine. Le sujet de requête de modèle pointe vers la dimension masquée nouvellement créée, qui reste compatible avec les rapports existants.

Les sujets de requête de modèle qui ne contiennent pas de niveaux ni de hiérarchies sont convertis en dimensions de mesure.

Après la mise à niveau, examinez le modèle afin de vérifier que les dimensions de mesure et les dimensions ordinaires nouvellement créées répondent à vos exigences en termes d'analyse et de génération de rapports.

Après la mise à niveau d'un modèle ReportNet 1.x vers un modèle IBM Cognos 8.3, les mesures qui ont été ajoutées au dossier des chiffres clés pour l'objet SAP BW sous-jacent après l'importation initiale ne sont pas ajoutées lors de la phase de synchronisation. La solution consiste à réimporter l'objet.

Ce problème n'affecte pas les modèles des versions ultérieures d'IBM Cognos BI.

---

## Vérification et réparation de modèles IBM Cognos ReportNet mis à niveau

Cette section et ses sous-sections s'appliquent uniquement lors de la mise à niveau de modèles depuis IBM Cognos ReportNet vers IBM Cognos Business Intelligence.

Après la mise à jour des métadonnées, IBM Cognos Framework Manager vous invite à vérifier le modèle. Des avertissements s'affichent pour chaque objet qui contient des informations de dimension ou l'un des types de données répertoriés dans la section concernant la mise à niveau et les types de données. Lorsque vous vérifiez des modèles de grande taille, contrôlez les espaces-noms un par un. Vous pouvez également vérifier un objet individuel.

Le processus de réparation évalue d'abord tous les éléments sélectionnés. Cette évaluation résout automatiquement les problèmes de nouveaux types de données et vous invite à réparer les informations de dimension dans le modèle. Vous pouvez sélectionner une ou plusieurs cases à cocher et réparer les éléments.

La vérification du modèle séparément du processus de mise à niveau offre de nombreux avantages. Cela vous permet :

- D'ouvrir un modèle existant, de le mettre à niveau et de le publier sans modification de la fonctionnalité de requête.
- De prendre le temps de réévaluer les exigences du modèle.
- De continuer de concevoir le modèle de production, de le publier sans les informations de dimension et de le faire évoluer progressivement vers des dimensions et des déterminants.

Lors du processus de mise à niveau, vous devez :

- Convertir des sujets de requête avec des informations de dimension en sujets de requête avec des déterminants ou des dimensions.
- Sélectionner et réparer des objets.

Si vous modifiez le modèle avant de le vérifier, ne modifiez pas les déterminants. Cela pourrait entraîner la perte des informations de dimension et vous empêcher de vérifier le modèle ultérieurement.

## Conversion d'informations de dimension en déterminants ou en dimensions

Dans IBM Cognos ReportNet, les informations de dimension combinaient l'unicité avec des hiérarchies dimensionnelles. Dans IBM Cognos BI, les informations de dimension pour les dimensions ordinaires sont scindées en déterminants et en hiérarchies. Les déterminants contrôlent l'unicité et la granularité. Ce contrôle est

requis pour les sujets de requête basés sur des sources de données relationnelles, notamment dans les requêtes à faits multiples et à granularité multiple. Les hiérarchies utilisent des concepts dimensionnels de hiérarchies, de niveaux, de clés et d'attributs pour toutes les sources de données.

Les sujets de requêtes avec des informations de dimension ne sont pas convertis en déterminants ou en dimensions lors de la mise à niveau. Les métadonnées que les informations de dimension indiquaient précédemment sont préservées dans le modèle et continuent d'exister pour les sujets de requête jusqu'à ce qu'ils soient réparés. Vous ne pouvez pas modifier les informations de dimension dans IBM Cognos BI, mais vous pouvez les mettre à niveau vers des déterminants ou des dimensions. Tant que vous n'avez pas mis à niveau les sujets de requête, IBM Cognos BI utilise les informations de dimension spécifiées précédemment dans ReportNet. Pour plus d'informations, voir «Mappage d'informations de dimension».

Si vous modifiez le modèle sans le vérifier après la mise à niveau, ne modifiez pas les déterminants. Cela peut entraîner la perte des informations de dimension. Lorsque vous vérifiez les informations de dimension, vous devez comprendre la façon dont elles sont appliquées au sujet de requête et comment vous utiliserez ce dernier dans un modèle dans IBM Cognos BI. Pour plus d'informations, voir «Vérification post-conversion», à la page 389.

Pour plus d'informations sur l'utilisation des métadonnées, voir Chapitre 9, «Guide de modélisation des métadonnées», à la page 327.

## **Mappage d'informations de dimension**

Vous pouvez mapper les informations de dimension contenues dans les sujets de requête de modèle ReportNet vers des sujets de requête avec des déterminants ou vers des dimensions contenues dans le modèle IBM Cognos BI.

- **Déterminants**

Utilisez des déterminants pour préserver les rapports existants lors de l'extension de votre application.

Les déterminants reflètent la granularité en représentant des sous-ensembles ou des groupes de données dans un sujet de requête. Ils permettent de garantir l'agrégation correcte de ces données répétées. Les déterminants sont importés en fonction d'informations de clé et d'index uniques dans la source de données.

Contrairement aux informations de dimension, les sujets de requête de modèle héritent des déterminants qui sont spécifiés pour les sujets de requête sous-jacents. Spécifiez les déterminants le plus tôt possible dans le modèle, généralement sur le sujet de requête de source de données. Lorsque vous spécifiez différents déterminants pour un contrôle de granularité différent, créez plusieurs instances du sujet de requête de source de données ou supprimez les déterminants de ce dernier. Créez ensuite de nouveaux déterminants sur chaque sujet de requête de modèle qui nécessite un contrôle de granularité différent.

- **Dimensions**

Utilisez des dimensions si vous utilisez votre modèle dans Analysis Studio, activez l'accès au détail vers le haut et vers le bas dans les rapports ou accédez aux fonctions membre dans les studios. Seules les dimensions ordinaires sont créées. Lors de la mise à niveau, il n'existe pas suffisamment de métadonnées pour créer des dimensions de mesure, par conséquent, vous devez les créer manuellement. Vous pouvez ensuite relier les dimensions aux mesures en définissant la portée du modèle.

Si vous mettez à niveau des sujets de requête vers des dimensions, les noms des éléments de requête contenus dans votre modèle sont modifiés. Par exemple, un élément de requête précédemment qualifié par *namespace.query.subject.query item* est désormais qualifié par *namespace.dimension.hierarchy.level.query item*. Etant donné que ce changement de nom peut rendre certains rapports non valides, mettez à niveau les sujets de requête vers les dimensions uniquement lorsque vous souhaitez utiliser un modèle existant pour créer une nouvelle application qui nécessite la fonction dimensionnelle.

Les informations de dimension sont mappées vers des déterminants et des dimensions, comme suit :

Informations de dimension	Déterminants	Dimensions
Hiérarchies	Pour les sources de données relationnelles, la première hiérarchie crée le déterminant.	Hiérarchies  Pour les sources de données SAP BW, les autres hiérarchies sont mises à niveau vers les autres hiérarchies d'une dimension.
Niveaux	Déterminants, identifiés de manière unique ou groupés par	Niveaux  Le premier niveau de la hiérarchie prend automatiquement la valeur Tout. Il contient un membre racine unique, lequel représente le niveau supérieur de la hiérarchie.  Vous ne pouvez pas supprimer ni déplacer le niveau Tout. Vous pouvez modifier son nom, sa description et son infobulle.
Clés	Si une clé unique n'est pas sélectionnée, les segments de clé des niveaux supérieurs sont inclus dans la clé.	Rôle _businessKey
	Si la clé unique est sélectionnée, seule la clé de segment, ou les segments, pour le niveau sont inclus dans la clé.	Niveau unique
Premier attribut de texte (par ordre alphabétique)		Rôle _memberCaption

Informations de dimension	Déterminants	Dimensions
Attributs	<p>Attributs</p> <p>Les attributs non associés sont affectés au dernier déterminant, lequel correspond généralement au niveau le plus bas.</p>	<p>Peuvent être affectés manuellement au rôle <code>_memberDescription</code>, à un rôle personnalisé ou à aucun rôle</p>

## Vérification post-conversion

Après avoir converti des sujets de requête avec des informations de dimension en des sujets de requête avec des déterminants, vous devez vérifier les paramètres suivants :

- Identification de façon unique  
Lorsque vous définissez un déterminant unique, vous indiquez que la clé contient suffisamment d'informations pour identifier un groupe au sein des données.
- Grouper par  
Vous devez définir **Grouper par** lorsque vous définissez un déterminant non unique. Celui-ci indique à IBM Cognos que, lorsque les clés ou attributs associés à ce déterminant sont répétés dans les données, il convient d'appliquer des fonctions d'agrégation et un groupement afin d'éviter le double comptage.
- Hiérarchies multiples ou secondaires qui existaient dans ReportNet  
Si deux hiérarchies existaient sur un sujet de requête dans ReportNet, seule la première hiérarchie est mise à niveau vers un déterminant pour les sources de données relationnelles. Vous devez créer un deuxième sujet de requête et indiquer manuellement les déterminants pour l'autre hiérarchie.  
Pour les sources de données SAP BW, les hiérarchies secondaires sont mises à niveau vers des déterminants.

Après avoir converti des sujets de requête avec des informations de dimension en dimensions, vous devez vérifier les paramètres suivants :

- Rôle `_businessKey`  
Ce rôle représente la clé pour le niveau et peut être affecté à un seul attribut dans un niveau.
- Niveau unique  
Un niveau unique indique que les clés des niveaux supérieurs ne sont pas nécessaires pour identifier les membres de ce niveau.
- Rôle `_memberCaption`  
Pour tirer parti des fonctions membres dans les studios IBM Cognos, vous devez affecter un rôle `_memberCaption` à chaque niveau d'une dimension. S'il n'existe aucun attribut pour le niveau, l'absence d'une légende est mise en évidence lorsque vous vérifiez le modèle.  
Toutes les légendes doivent avoir le type de données chaîne. Si aucun attribut de ce type n'est disponible, créez un calcul de type de données chaîne et affectez le rôle `_memberCaption` au nouvel élément. Ceci est avant tout un problème pour Analysis Studio.
- Attributs

Incluez des attributs de la dimension et associez-les au niveau correct. Par défaut, ils sont inclus sans rôle. Vous pouvez créer des rôles personnalisés ou affecter des attributs à des rôles existants.

- Hiérarchies multiples

Une dimension ordinaire peut comporter plusieurs hiérarchies ; néanmoins, vous ne pouvez utiliser qu'une seule hiérarchie à la fois dans une requête. Par exemple, vous ne pouvez pas utiliser une hiérarchie sur les lignes d'un rapport de type tableau croisé et une autre hiérarchie de la même dimension dans les colonnes. Si vous avez besoin de ces deux hiérarchies dans le même rapport, vous devez créer deux dimensions, une pour chaque hiérarchie.

## Sélection et réparation d'objets dans les modèles IBM Cognos ReportNet mis à niveau

Le processus de réparation évalue d'abord tous les éléments sélectionnés. Cette évaluation résout automatiquement les problèmes de nouveaux types de données et vous invite à réparer les informations de dimension dans le modèle.

Les avertissements suivants peuvent s'afficher lorsque vous vérifiez et réparez un modèle mis à niveau :

Avertissement	Description
Réévaluation requise	<p>Ce message est le plus souvent lié à des modifications de type de données.</p> <p>Vous pouvez sélectionner et réparer la plupart des éléments associés à cet avertissement. Utilisez l'option de réparation pour évaluer et mettre à niveau des métadonnées spécifiques.</p>
L'expression de jointure entre en conflit avec les informations sur le déterminant du sujet de requête	<p>Parfois, les informations d'index et de clé définies pour un sujet de requête impliquent un niveau de granularité qui ne correspond pas aux relations spécifiées sur un sujet de requête.</p>
Aucun des éléments de requête de ce niveau n'a le rôle de Légende	<p>Lorsque vous définissez des niveaux, vérifiez que des rôles de clé métier et de légende sont spécifiés. Ces rôles sont requis pour les fonctions membres des studios IBM Cognos et l'arborescence orientée membre d'Analysis Studio.</p> <p>Toutes les légendes doivent avoir le type de données chaîne. Si aucun attribut de ce type n'est disponible, créez un calcul de type de données chaîne et affectez le rôle de légende du membre au nouvel élément. Ceci est avant tout un problème pour Analysis Studio.</p>



Avertissement	Description
Un ou plusieurs déterminants décrivant les clés et attributs du sujet de requête doivent être spécifiés	Les déterminants sont basés sur des informations de clé de la source de données. Des déterminants peuvent ne pas exister pour un sujet de requête mis à niveau depuis ReportNet, en particulier pour un sujet de requête de modèle. Utilisez des déterminants pour spécifier la granularité des données du sujet de requête et les dépendances fonctionnelles entre les éléments de requête. Toutefois, il n'est pas obligatoire de spécifier des déterminants pour les sujets de requête qui représentent un niveau unique ou des données de fait.

---

## Mise à niveau de projets segmentés et liés

Ces informations s'appliquent quelle que soit la version du produit à partir de laquelle vous effectuez la mise à niveau.

Pour les projets segmentés, les segments sont automatiquement mis à niveau lorsque vous ouvrez le projet principal.

Vous devez mettre à niveau un projet lié avant de mettre à niveau tous les projets qui l'utilisent.

Si vous disposez d'un modèle segmenté ou lié créé avec une version précédente d'IBM Cognos Framework Manager, vous devez vérifier les paramètres de restriction pour chaque modèle enfant et parent avant et après la mise à niveau. Les paramètres de restriction pour le modèle parent de niveau supérieur sont appliqués lors de la publication.

Si vous exécutez un script créé avec une version précédente de Framework Manager, le script ignore le processus de mise à niveau. Vérifiez que les paramètres de restriction sont corrects avant et après l'exécution du script.

Si un projet segmenté n'est pas automatiquement mise à niveau, procédez comme suit.

### Procédure

1. Ouvrez chaque segment en tant que projet distinct en commençant par le segment de plus bas niveau dans la hiérarchie.
2. Exécutez les étapes de mise à niveau du modèle.  
Pour plus d'informations, voir Chapitre 11, «Mise à niveau de modèles», à la page 383.  
Ne réparez pas le segment.
3. Lorsque la mise à niveau est terminée, enregistrez le projet.
4. Mettez à niveau chaque segment dans la hiérarchie en revenant dans le projet principal.
5. Réparez la totalité du modèle dans le projet principal.
6. Après avoir mis à niveau le projet principal, restituez chaque segment, puis restituez le projet principal.



---

## Annexe A. Fonctions d'accessibilité

Les fonctions d'accessibilité permettent aux utilisateurs souffrant d'un handicap physique, comme une mobilité réduite ou une vision limitée, d'utiliser avec succès les produits informatiques.

---

### Fonctions d'accessibilité dans Framework Manager

Les principales fonctions d'accessibilité sont les raccourcis clavier et les touches de commande que vous pouvez utiliser pour naviguer dans Framework Manager.

- Une lettre soulignée à l'écran désigne un raccourci clavier ; par exemple, F est le raccourci clavier du menu Fichier. Dans Microsoft Windows, appuyez sur la touche Alt, puis sur le raccourci clavier pour déclencher une action. Par exemple, Alt+F permet d'afficher le menu Fichier. S'ils sont activés, vous pouvez également utiliser les raccourcis-clavier étendus.
- Les touches de commande déclenchent directement une action et s'utilisent généralement avec les touches Ctrl. Par exemple, pour imprimer, appuyez sur Ctrl+P.

---

### Raccourcis-clavier pour Framework Manager

Vous pouvez utiliser des raccourcis-clavier pour naviguer et exécuter certaines tâches dans Framework Manager.

Ce produit utilise des touches de navigation Microsoft Windows standard en plus des touches propres aux applications.

Les raccourcis-clavier suivants sont basés sur les claviers standard américains.

Description	Raccourci
Aide contextuelle	F1
Nouveau fichier	Ctrl + N
Ouvrir un fichier	Ctrl + O
Enregistrer le fichier	Ctrl + S
Annuler	Ctrl + Z
Rétablir	Ctrl + Y
Couper	Ctrl + X
Copier	Ctrl + C
Coller	Ctrl + V
Supprimer	Suppr

### Raccourcis-clavier pour Model Design Accelerator

Vous pouvez utiliser des raccourcis-clavier pour naviguer et exécuter certaines tâches dans Model Design Accelerator.

Description	Raccourci de diagramme
Présentation automatique en tant qu'arborescence	Ctrl + A
Présentation en tant qu'étoile	Ctrl + S
Zoom avant	Ctrl + +
Zoom arrière	Ctrl + -
Adapter le diagramme à la fenêtre	Ctrl + 0 (zéro)
Zoomer à 100 %	Ctrl + 1 (un)
Annuler	Ctrl + Z
Rétablir	Ctrl + Y
Ajouter des tables directement liées	Ctrl + D
Ajouter toutes les tables liées par des relations Plusieurs à un	Ctrl + T
Ajouter uniquement des tables avec des relations Un à plusieurs	Ctrl + O
Ajouter uniquement des tables avec des relations Plusieurs à un	Ctrl + M
Retirer les tables sélectionnées	Suppr
Retirer toutes les tables sauf celles sélectionnées	Maj + Suppr

Description	Raccourci Model Accelerator
Renommer	F2
Supprimer	Suppr
Zoom avant	Ctrl + +
Zoom arrière	Ctrl + -
Adapter le diagramme à la fenêtre	Ctrl + 0 (zéro)
Zoomer à 100 %	Ctrl + 1 (un)
Annuler	Ctrl + Z
Rétablir	Ctrl + Y

---

## IBM et l'accessibilité

Pour en savoir plus sur l'engagement d'IBM en matière d'accessibilité, voir le centre IBM Accessibility Center (<http://ibm.com/able>).

---

## Annexe B. Traitement des incidents

Lorsque vous travaillez avec Framework Manager, il est possible que vous rencontriez certains problèmes.

Pour trouver d'autres rubrique sur le traitement des incidents, voir le *Guide d'administration et de sécurité* IBM Cognos.

---

### Impossible de comparer deux objets CLOB dans Oracle

Si vous utilisez Oracle et que vous demandez à IBM Cognos Business Intelligence de comparer deux objets CLOB (par exemple, dans une expression telle que C2 = C3), une erreur d'exécution Oracle se produit.

Pour éviter cela, utilisez la méthode DBMS\_LOB.compare :

```
where 0 = dbms_lob.compare (c1, c2)
```

---

### Erreur de mémoire insuffisante avec des métadonnées importées ERWin

Lorsque vous testez les sujets de requête basés sur une table d'affichage, une erreur de mémoire insuffisante peut se produire lors de l'exécution de l'opération `sqlPrepareWithOptions`.

La solution consiste à créer un sujet de requête de source de données à l'aide de la même connexion au Content Manager que le modèle ERWin.

---

### Framework Manager ne peut pas accéder à l'URI de la passerelle

Lorsque vous créez un projet dans Framework Manager, Framework Manager ne parvient pas à accéder à l'URI de la passerelle.

Le message d'erreur suivant s'affiche :

Impossible d'accéder au service à l'adresse URL suivante :

```
http://nomôte:80/ibmcognos/cgi-bin/cognos.cgi/  
b_acton=xts.run&m=portal/close.xts
```

Vérifiez si les informations sur l'URI de la passerelle sont configurées correctement et si le service est disponible.

Pour en savoir davantage, veuillez vous adresser à l'administrateur.

Ce message s'affiche si la passerelle n'est pas configurée correctement. L'URI de la passerelle doit être défini sur le nom de l'ordinateur sur lequel IBM Cognos BI est installé et doit correspondre au type de passerelle que vous utilisez. Vous devez vous connecter en tant qu'administrateur pour configurer l'URI de la passerelle.

#### Procédure

1. Fermez Framework Manager.

2. Dans IBM Cognos Configuration, dans la fenêtre **Explorateur**, cliquez sur **Environnement**.
3. Dans la zone **URI de la passerelle** de la fenêtre **Propriétés**, saisissez la valeur appropriée :
  - Pour utiliser ISAPI, remplacez cognos.cgi par **cognosisapi.dll**.
  - Pour utiliser apache\_mod, remplacez cognos.cgi par **mod\_cognos.dll**.
  - Pour utiliser une passerelle de type servlet, saisissez la commande suivante :  
`http[s]://host:port/context_name/servlet/Gateway`  
**Remarque** : *context\_name* est le nom de l'application Web de passerelle de type servlet que vous avez défini lors du déploiement du fichier WAR de la passerelle de type servlet.
  - Si vous n'utilisez pas de serveur Web, saisissez la syntaxe suivante pour utiliser le répartiteur en tant que passerelle :  
`http[s]://host:port/p2pd/servlet/dispatch`
4. Si nécessaire, remplacez la portion de nom d'hôte localhost de l'**URI de la passerelle** par l'adresse IP ou le nom de l'ordinateur.
5. Dans le menu **Fichier**, cliquez sur **Enregistrer**.
6. Dans le menu **Actions**, cliquez sur l'option **Redémarrer**.

---

## Les noms d'objets s'affichent dans une langue erronée

Lorsque vous importez plusieurs langues à partir d'une requête SAP BW vers un modèle Framework Manager, tous les noms d'objets récupérés depuis SAP BW ne s'affichent pas dans la langue appropriée.

Pour éviter ce problème, enregistrez à nouveau la requête SAP BW pour chaque langue de connexion dans Business Explorer Query Designer. Le texte s'affiche ensuite dans la langue appropriée dans Framework Manager.

---

## Dans Oracle, des jointures externes complètes renvoient des résultats incorrects

Lorsque vous utilisez une source de données Oracle antérieure à la version 10.2, des jointures externes complètes renvoient des résultats de données incorrects. Pour éviter cela, IBM Cognos BI traite ces jointures comme des opérations locales.

Suite à ce traitement, vous devez attribuer la valeur `limitedLocal` (traitement local limité) au paramètre de traitement des requêtes pour tout projet dans lequel des jointures externes implicites ou explicites sont prévues.

---

## Erreur lors du test des sujets de requête dans un modèle importé à partir de Teradata

Vous utilisez un modèle importé à partir de Teradata. Le message d'erreur suivant s'affiche lorsque vous testez certains sujets de requête contenant des éléments graphiques :

QE-DEF-0177 Une erreur s'est produite lors de l'exécution de l'opération 'sqlOpenResult' status='-28'.

UDA-SQL-0114 Le curseur fourni pour l'opération "sqlOpenResult" est inactif.

UDA-SQL-0107 Une exception s'est produite pendant l'opération "SgiCursor::doOpenResult()".

[NCR][ODBC Teradata Driver][Teradata RDBMS] La traduction d'une chaîne de caractères inconnue a été demandée.

Cela est dû au fait que les types de données GRAPHIC et VARGRAPHIC ne sont pas pris en charge.

---

## Erreur liée au sujet de requête SQL en saisie directe

Définissez ce qui suit dans DB2 :

```
create type address as (  
  number character (6),  
  street varchar(35),  
  city varchar(35)  
)  
MODE DB2SQL;  
create table emp ( emp_no int, emp_address address);  
Select e.emp_no, e.emp_address..street from emp e  
SQL0206N "aBmtQuerySubject.2
```

Lorsque vous définissez un sujet de requête SQL en saisie directe, une erreur se produit à cause du nom attribué à la référence d'attribut dans le type structuré.

Pour résoudre ce problème, procédez de l'une des deux façons suivantes :

- affectez un nom de corrélation simple à la colonne dans le sujet de requête d'origine, par exemple `Select e.emp_no, e.emp_address..street as "ABC" from emp e ;`
- utilisez la notation passe-système pour le sujet de requête en plaçant la colonne entre accolades doubles (`{{ }}`).

---

## Erreur QE-DEF-0259

Cette erreur se produit si vous placez des accolades { } aux mauvais endroits dans une expression. IBM Cognos BI suppose que tout ce qui est placé entre accolades se situe au même niveau que la fonction. Si vous avez utilisé des accolades aux mauvais endroits dans une expression, le message suivant s'affiche :

QE-DEF-0259 Une erreur d'analyse syntaxique s'est produite

Vous pouvez également utiliser des accolades pour envoyer une syntaxe unique à la source de données. Par exemple, votre base de données utilise un mot clé donné pour une fonction, mais celui-ci n'est pas utilisé dans IBM Cognos BI.

IBM Cognos BI ne valide pas la syntaxe que vous saisissez entre des accolades. La syntaxe est envoyée directement à la base de données.

La solution consiste à s'assurer que les accolades ne sont pas placées aux mauvais endroits.

Par exemple, vous saisissez ce qui suit dans une expression :

```
[ss_ole_both].[authors_lith].[au_id] = [ss_ole_both].[authors_latin].[au_id]  
{ collate Lithuanian_CI_AI}
```

Vous obtenez le message d'erreur suivant :

QE-DEF-0259 Une erreur d'analyse syntaxique s'est produite avant ou près de la position : 75, le texte commence à la position : 5  
"le\_both].[authors\_lith].[au\_id]=[ss\_ole\_both].[authors\_latin].[au\_id]{"

L'expression correcte est la suivante :

```
{ Q3.au_id } = { Q4.au_id collate lithuanian_CI_AI  
}
```

---

## La dimension des données clés externalisées conserve une ancienne valeur d'invite

Vous disposez d'une dimension de chiffres clés (SAP BW) qui contient une invite facultative. Si vous externalisez cette dimension sous forme de fichier .csv ou .tab, le fichier externalisé ne contient pas toutes les lignes de données. Cela est dû au fait que la valeur d'invite est conservée.

Par exemple, vous définissez la valeur d'invite pour la dimension lorsque vous testez celle-ci dans Framework Manager. La valeur d'invite est conservée dans le cache. Même si vous effacez la valeur de l'invite dans la boîte de dialogue Invite, l'externalisation de la dimension des chiffres clés génère un fichier contenant des données qui sont filtrées par la dernière invite utilisée.

Pour éviter ce problème, procédez de l'une des façons suivantes :

- Ne testez pas la dimension de chiffres clés avant de l'externaliser.
- Fermez le modèle, rouvrez-le, puis externalisez la dimension de chiffres clés.

---

## Des modèles anciens affichent un niveau de sécurité pour des objets

Si vous utilisez un modèle IBM Cognos créé précédemment, un niveau de sécurité a pu être défini pour des objets. Or, cette option n'est pas prise en charge.

La solution consiste à vérifier et à réparer l'ancien modèle avant de le publier.

### Procédure

1. Dans le menu **Projet**, cliquez sur **Vérifier le modèle**.
2. Sélectionnez la vue de sécurité qui fait référence à un niveau et cliquez sur **Réparer**.

---

## Echec de l'exportation d'un modèle Framework Manager vers un fichier CWM avec l'erreur Impossible de trouver MILOG.TXT

L'exportation d'un modèle Framework vers un fichier CWM échoue avec l'erreur *Impossible de trouver MILOG.TXT* lorsque le chemin contient des caractères japonais.

Pour résoudre ce problème, procédez de l'une des façons suivantes :

- Indiquez un chemin d'exportation qui n'utilise pas de caractères japonais.
- Remplacez la langue par défaut du système de votre ordinateur par le japonais. Vous pouvez définir la langue par défaut du système dans le Panneau de configuration, sous **Options régionales et linguistiques** -> **Options avancées**. Pour plus d'informations sur le mode opérationnel à utiliser, consultez l'aide du système d'exploitation Microsoft Windows.



---

## Différence de code SQL pour les jointures internes après une mise à niveau vers IBM Cognos BI version 8.3 et versions ultérieures

En cas de migration depuis une version antérieure à la version 8.3 du produit, des différences peuvent être constatées lors de la création de code SQL utilisée pour la syntaxe INNER JOIN.

Vous pouvez contrôler la syntaxe SQL utilisée pour les jointures internes en configurant le paramètre **Syntaxe de jointure SQL**. La syntaxe de jointure SQL créée pour toutes les versions d'IBM Cognos BI produit le même résultat.

Si vous utilisez la technologie de matérialisation RDBMS, qui peut être mise en oeuvre à l'aide d'une syntaxe implicite ou explicite, vous devez veiller à sélectionner le même paramètre de syntaxe pour la restriction **Syntaxe de jointure SQL** de votre modèle.

---

## Jointures externes complètes non envoyées à Oracle 9i et 10GR1

Par défaut, IBM Cognos BI n'envoie pas de jointure externe complète à ORACLE 9i et 10GR1 en raison du bogue Oracle n° 2874433. Cela nécessite l'utilisation d'un traitement local limité dans IBM Cognos BI.

Pour activer les jointures externes complètes avec Oracle, vous devez :

- vérifier que vous disposez des ensembles de correctifs requis, notamment le correctif concernant le bogue n° 2874433 ;
- modifier le fichier cogdmor.ini afin d'activer les jointures externes complètes (Full\_outer\_join=T).

Etant donné que les éditions manuelles des paramètres ini sont écrasées par l'installation suivante, vous devez les répliquer manuellement sur toutes les machines où vous avez installé IBM Cognos BI ou Framework Manager.

---

## Différences inexplicables dans les calculs de nombres

Vous risquez de trouver des différences dans les calculs de nombres en raison d'erreurs d'arrondi.

Par exemple :

- En exécutant des tests de régression, vous trouvez des variations dans les nombres. L'écart provient uniquement de l'arrondi des décimales.
- Vous choisissez de ne pas afficher les zéros dans les rapports, cependant ils s'affichent tout de même, car des décimales (par exemple : 0,00000000000000426) sont arrondies à zéro dans les rapports.

Les problèmes d'arrondi ne sont pas spécifiques au logiciel IBM Cognos. Ils peuvent survenir dans n'importe quel environnement où des arrondis sont effectués.

### Erreurs d'arrondi binaire

Les différences dans les calculs peuvent se produire en raison d'erreurs d'arrondi binaire. Par exemple, si le nombre 1,1 est représenté par un nombre en virgule

flottante binaire et le format de votre rapport inclut un grand nombre de nombres décimaux, le nombre 1,1 peut en réalité représenter un nombre tel que 1,09999999999997.

Si votre rapport est formaté pour utiliser uniquement une décimale après la virgule flottante, l'arrondi de décimales est utilisé, en guise de compensation à l'arrondi binaire. Le nombre affiché est ainsi 1,1, alors qu'en réalité il s'agit de 1,09999999999997. Lorsque ce nombre est utilisé dans des calculs, vous risquez de trouver des erreurs d'arrondi. Par exemple, les calculs de Microsoft Excel utilisent des nombres binaires (sans arrondir les nombres décimaux) ; cependant, le formatage des rapports affiche des décimales arrondies, ce qui peut générer de légères différences.

## Erreurs d'arrondi de division

Les calculs qui comportent une division impliquent généralement des erreurs d'arrondi, sans tenir compte du mode de représentation des nombres. Les exemples qui illustrent ce type de calcul sont la Moyenne et le Pourcentage de la base.

## Instructions de conception pour réduire l'effet de l'arrondi

La meilleure solution consiste à modifier le schéma ou le modèle de cube de la base de données sous-jacente. Cependant, cette opération n'est pas toujours possible. Une solution alternative consiste à réduire l'effet de l'arrondi. Pour ce faire, suivez les instructions suivantes lors de la création de rapports, de modèles dans IBM Cognos Framework Manager et de cubes OLAP externes :

- Évitez de stocker des données avec des virgules flottantes autant que possible. Cela s'applique particulièrement pour les valeurs monétaires, qui doivent être stockées sous forme de décimales à point fixe ou d'entiers avec une valeur d'échelle telle que 2.

Par exemple, dans un cube, les revenus pour le Matériel de camping en 2012 sont de 20 471 328,88 \$. Si les détails des revenus sont stockés sous forme de nombres en virgule flottante, des erreurs d'arrondi risquent de se produire lors du calcul des revenus.

Les erreurs d'arrondi peuvent présenter de légères différences, en fonction de l'ordre du calcul. Si les revenus pour les Produits sont calculés en premier et ensuite ceux associés aux Heures, vous risquez de trouver une erreur d'arrondi différente par rapport aux revenus associés aux Heures calculés avant ceux des Produits.

Les revenus totaux peuvent être calculés comme le nombre ci-dessus. De légères différences peuvent se produire, par exemple, 20 471 328,8800001 \$ par opposition à 20 471 328,88 \$. Le nombre interne peut être légèrement différent de celui affiché. Ce nombre peut même être utilisé pour différentes exécutions du même rapport, selon l'ordre utilisé par le moteur OLAP pour le calcul.

- Dans les rapports, évitez la division autant que possible. Lorsque vous ne pouvez éviter une division, essayez de l'effectuer aussi tardivement que possible dans le processus de calcul. Par exemple, au lieu d'effectuer l'opération `Total([Revenus]/1000)`, utilisez `Total([Revenus])/1000`.
- Lors de comparaisons, ajoutez une marge pour vous permettre d'arrondir. Par exemple, vous pouvez faire de [% de profit] une valeur fractionnaire mise en forme en tant que pourcentage sans décimales. Cependant, le filtre [% de profit]<>0 (ou [% de profit] NON COMPRIS ENTRE 0 et 0) rejette les valeurs zéro et peut renvoyer des valeurs qui s'affichent comme 0 % après le formatage. Pour éviter cela, effectuez le filtrage de l'une des manières suivantes :

- [% de profit] NON COMPRIS ENTRE -0,005 et 0,005
- ([Profit %] <- 0.005) OR ([Profit %]> 0.005)

Notez que 0,005 équivaut à 0,5 %. Ce pourcentage s'affiche en tant que 0 % ou 1 % en fonction des pertes de précision avec la virgule flottante.

Dans certains cas, vous pouvez contrôler les erreurs d'arrondi en arrondissant les valeurs de manière explicite. Par exemple, au lieu de l'opération [% de profit], utilisez `arrondi([% de profit],2)`.

- Recalculez systématiquement les nombres au lieu de réutiliser des calculs qui peuvent contenir des décimales arrondies.

Il existe quelques points pouvant être pris en compte pour Microsoft Analysis Services 2005/2008, particulièrement lors de la comparaison de résultats de rapports issus de différentes exécutions (comme dans Lifecycle Manager). Pour en savoir davantage, reportez-vous à la documentation Microsoft.

---

## Recherche de valeurs pouvant renvoyer des résultats inattendus

Dans l'éditeur d'expression, lorsque vous recherchez des valeurs pour une donnée élémentaire, les résultats que vous obtenez peuvent contenir des résultats inattendus si la donnée élémentaire ne correspond pas à une donnée de type chaîne. Etant donné que les utilisateurs peuvent éditer l'expression d'une donnée élémentaire, IBM Cognos Business Intelligence ne peut pas déterminer le type de données avec certitude.

Par conséquent, IBM Cognos Business Intelligence déduit le type de données de la donnée élémentaire en examinant son ensemble d'agrégats et d'agrégats de cumul.



---

## Annexe C. Utilisation de l'éditeur d'expression

Une expression est une combinaison d'opérateurs, de constantes, de fonctions et d'autres composants dont il découle une valeur unique. Les expressions servent à créer des calculs et des définitions de filtre. Un calcul est une expression utilisée par vous pour créer une nouvelle valeur à partir de valeurs existantes contenues dans une donnée élémentaire. Un filtre est une expression qui vous sert à extraire un ensemble spécifique d'enregistrements.

Construisez des expressions qui utilisent les composants suivants.

---

### Opérateurs

Un opérateur indique quel traitement effectuer sur les valeurs à sa gauche et à sa droite. Les opérateurs sont semblables à des fonctions, dans le sens qu'ils traitent des données élémentaires et transmettent un résultat.

(

Identifie le début d'une expression.

#### Syntaxe

( expression )

)

Identifie la fin d'une expression.

#### Syntaxe

( expression )

\*

Multiplie deux valeurs numériques.

#### Syntaxe

value1 \* value2

,

Sépare les différents éléments de l'expression.

#### Syntaxe

expression ( parameter1, parameter2 )

/

Divise deux valeurs numériques.

#### Syntaxe

value1 / value2

||

Concatène ou assemble deux chaînes.

## Syntaxe

string1 || string2

**+**

Additionne deux valeurs numériques.

## Syntaxe

value1 + value2

**-**

Soustrait deux valeurs numériques ou rend une valeur numérique négative.

## Syntaxe

value1 - value2  
or  
- value

**<**

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs inférieures à "value2".

## Syntaxe

value1 < value2

**<=**

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs inférieures ou égales à "value2".

## Syntaxe

value1 <= value2

**<>**

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs différentes de "value2".

## Syntaxe

value1 <> value2

**=**

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs égales à "value2".

## Syntaxe

value1 = value2

**>**

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs supérieures à "value2".

## Syntaxe

value1 > value2

->

Sépare les composants d'une expression de membre littérale.

### Syntaxe

[namespace].[dimension].[hierarchy].[level]->[L1]

>=

Compare les valeurs représentées par "value1" et "value2" et extrait les valeurs supérieures ou égales à "value2".

### Syntaxe

value1 >= value2

et

Renvoie la valeur "true" si les conditions des deux côtés de l'expression sont vraies.

### Syntaxe

argument1 and argument2

auto

Fonctionne avec des expressions de récapitulatif pour définir la portée à ajuster en fonction des colonnes de groupement dans la requête. La portée dépend du contexte.

### Syntaxe

aggregate\_function ( expression AUTO )

between

Détermine si une valeur se situe à l'intérieur d'une plage donnée.

### Syntaxe

expression between value1 and value2

### Exemple

[Revenue] between 200 and 300

Résultat

Renvoie le nombre de résultats correspondant aux revenus compris entre 200 et 300.

Données du résultat

Revenue	Between
\$332.06	false
\$230.55	true
\$107.94	false

case

Fonctionne avec When, Then, Else et End. "Case" identifie le début d'une situation spécifique dans laquelle des actions When, Then et Else sont définies.

## Syntaxe

```
case expression { when expression then expression } [ else  
expression ] end
```

## contains

Détermine si "chaîne1" contient "chaîne2".

## Syntaxe

```
string1 contains string2
```

## currentMeasure

Mot clé qui peut être utilisé comme premier argument dans les fonctions récapitulatives de membre. Cette fonction apparaît dans l'exemple de rapport Revenus totaux par pays, accessible dans le module Entrepôt de données VA (requête).

## Syntaxe

```
aggregate_function ( currentMeasure within set expression )
```

## default

Fonctionne avec la syntaxe lookup.

## Syntaxe

```
lookup (....) in (....) default (....)
```

## distinct

Mot clé utilisé dans une expression agrégée afin d'inclure uniquement les occurrences distinctes des valeurs. Voir aussi la fonction unique.

## Syntaxe

```
distinct dataItem
```

## Exemple

```
count ( distinct [OrderDetailQuantity] )
```

Résultat

1704

## else

Fonctionne avec la syntaxe if ou case. Si la condition if ou l'expression case n'ont pas pour valeur True, l'expression else est utilisée. Cette fonction apparaît dans l'exemple de rapport Les 10 meilleurs détaillants de 2005, accessible dans le module Entrepôt de données VA (analyse).

## Syntaxe

```
if ( condition ) then .... else ( expression ) , or case .... else ( expression ) end
```

## end

Indique la fin d'une syntaxe case ou when.



## Syntaxe

```
case .... end
```

## ends with

Détermine si "chaîne1" se termine par "chaîne2".

## Syntaxe

```
string1 ends with string2
```

## escape

Détermine si "string1" correspond à la structure de "string2", avec le caractère "char" utilisé de manière facultative en caractère d'échappement dans la chaîne de modèle.

## Syntaxe

```
string1 LIKE string2 [ ESCAPE char ]
```

## Exemple 1

```
[PRODUCT_LINE] like 'G%'
```

Résultat

Toutes les lignes de produit commençant par "G".

## Exemple 2

```
[PRODUCT_LINE] like '%Ga%' escape 'a'
```

Résultat

Toutes les lignes de produit se terminant par "G%".

## for

Fonctionne avec des expressions de récapitulatif pour définir la portée de l'agrégation dans la requête.

## Syntaxe

```
aggregate_function ( expression for expression { , expression } )
```

## for all

Fonctionne avec des expressions de récapitulatif pour ajuster la portée à toutes les colonnes de groupement dans la requête. Voir aussi la clause for.

## Syntaxe

```
aggregate_function ( expression for ALL expression { , expression } )
```

## for any

Fonctionne avec des expressions de récapitulatif pour ajuster la portée en fonction d'un sous-ensemble des colonnes de groupement dans la requête. Equivalent à la clause for.

## Syntaxe

```
aggregate_function ( expression for ANY expression { , expression } )
```

## for report

Fonctionne avec des expressions de récapitulatif pour définir la portée à l'ensemble de la requête. Voir aussi la clause for. Cette fonction apparaît dans l'exemple de rapport Retours et satisfaction des clients, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
aggregate_function ( expression for report )
```

## if

Fonctionne avec les syntaxes then et else. If définit une condition ; lorsque cette condition a pour valeur True, l'expression then est utilisée. Lorsque la condition if n'a pas pour valeur True, l'expression else est utilisée. Cette fonction apparaît dans l'exemple de rapport Les 10 meilleurs détaillants de 2005, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
if ( condition ) then ( expression ) else ( expression )
```

## in

Détermine si "expression1" se trouve dans une liste d'expressions donnée.

### Syntaxe

```
expression1 in ( expression_list )
```

## in\_range

Détermine si "expression1" existe dans une liste de valeurs constantes ou de plages donnée.

### Syntaxe

```
expression1 in_range { constant : constant [ , constant : constant ] }
```

### Exemple 1

```
[code] in_range { 5 }
```

Résultat

Equivalent à [code] = 5.

### Exemple 2

```
[code] in_range { 5: }
```

Résultat

Equivalent à [code] >= 5.

### Exemple 3

```
[code] in_range { :5 }
```

Résultat

Equivalent à [code] <= 5.

### Exemple 4

```
[code] in_range { 5:10 }
```

Résultat

Equivalent à ( [code] >= 5 et [code] <= 10 ).

### Exemple 5

```
[code] in_range { :5,10,20: }
```

Résultat

Equivalent à ( [code] <= 5 ou [code] = 10 ou [code] >= 20 ).

## is missing

Détermine si "valeur" ne se trouve pas dans les données.

### Syntaxe

```
value is missing
```

## is null

Détermine si "valeur" ne se trouve pas dans les données.

### Syntaxe

```
value is null
```

## is not missing

Détermine si "valeur" se trouve dans les données.

### Syntaxe

```
value is not missing
```

## is not null

Détermine si "valeur" se trouve dans les données.

### Syntaxe

```
value is not null
```

## like

Détermine si "string1" correspond à la structure de "string2", avec le caractère "char" utilisé de manière facultative en caractère d'échappement dans la chaîne de modèle.

### Syntaxe

```
string1 LIKE string2 [ ESCAPE char ]
```

### Exemple 1

```
[PRODUCT_LINE] like 'G%'
```

Résultat

Toutes les lignes de produit commençant par "G".

## Exemple 2

```
[PRODUCT_LINE] like '%G%' escape 'a'
```

Résultat

Toutes les lignes de produit se terminant par "G%".

## lookup

Recherche les données indiquées et les remplace par la valeur spécifiée. Il est préférable d'utiliser la syntaxe "case".

### Syntaxe

```
lookup ( name ) in ( value1 --> value2 ) default ( expression )
```

### Exemple

```
lookup ( [Country] ) in ( 'Canada'--> ( [List Price] * 0.60 ),  
'Australia'--> ( [List Price] * 0.80 ) ) default ( [List Price] )
```

## not

Renvoie TRUE si "argument" est faux ou FALSE si "argument" est vrai.

### Syntaxe

```
NOT argument
```

## or

Renvoie TRUE si "argument1" ou "argument2" est vrai.

### Syntaxe

```
argument1 or argument2
```

## prefilter

Effectue un calcul récapitulatif avant d'appliquer le filtre récapitulatif.

### Syntaxe

```
summary_function ([expression] prefilter)
```

### Exemple

```
total ( [Quantity] for report prefilter )  
summaryFilter: total(  
  [Quantity] for [ProductNo] ) > 50000
```

Résultat

Additionne les quantités dans un rapport avant l'application du filtre récapitulatif.

Données du résultat

<b>PNo</b>	<b>Total</b>	<b>Total</b>	<b>Total</b>
-----	-----	-----	-----
88	54928	298140	2215354
89	51126	298140	2215354
90	69996	298140	2215354
94	69004	298140	2215354

<b>PNo</b>	<b>Total</b>	<b>Total</b>	<b>Total</b>
95	53086	298140	2215354

## ROWS

Compte le nombre de lignes dans le résultat de la requête. S'utilise conjointement avec Count ().

### Syntaxe

```
count ( ROWS )
```

## ROWS

Compte le nombre de lignes dans le résultat de la requête. S'utilise conjointement avec Count ().

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
count ( ROWS [ expression ] )
```

## starts with

Détermine si "chaîne1" débute par "chaîne2".

### Syntaxe

```
string1 starts with string2
```

## then

Fonctionne avec la syntaxe if ou case. Lorsque la condition if ou l'expression when ont pour valeur True, l'expression then est utilisée. Cette fonction apparaît dans l'exemple de rapport Les 10 meilleurs détaillants de 2005, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
if ( condition ) then ..., or case expression when expression  
then .... end
```

## when

Fonctionne avec la syntaxe Case. Vous pouvez définir les conditions exercées lorsque l'expression WHEN vaut True.

### Syntaxe

```
case [expression] when ... end
```

---

## Récapitulatifs

Cette liste contient des fonctions prédéfinies qui renvoient soit une valeur récapitulative unique pour un groupe de valeurs liées, soit une valeur récapitulative différente pour chaque instance d'un groupe de valeurs liées.

## Fonctions statistiques

Les fonctions récapitulatives contenues dans cette liste sont de nature statistique.

## corr

Renvoie le coefficient de corrélation d'un ensemble de paires de nombres. Le calcul est le suivant :  $\text{COVAR\_POP}(\text{numeric\_expression1}, \text{numeric\_expression2}) / (\text{STDDEV\_POP}(\text{numeric\_expression1}) * \text{STDDEV\_POP}(\text{numeric\_expression2}))$

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
corr ( numeric_expression1 , numeric_expression2 [ auto ] )  
corr ( numeric_expression1 , numeric_expression2 for [ all|any ]  
expression { , expression } )  
corr ( numeric_expression1 , numeric_expression2 for report )
```

### Exemple

```
corr ( Cost , Margin for report)
```

Résultat

Le coefficient de corrélation entre Cost et Margin.

Données du résultat

Cost	Margin	corr (Cost, Margin for report)
4	0.33	0.0872648
5	0.28	0.0872648
9.22	0.23	0.0872648
15.93	0.28	0.0872648
34.97	0.3	0.0872648

## covariance-pop

Renvoie la covariance de population d'un ensemble de paires de nombres.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
covariance-pop ( numeric_expression1 , numeric_expression2 )  
covariance-pop ( numeric_expression1 , numeric_expression2 for  
[ all|any ] expression { , expression } )  
covariance-pop ( numeric_expression1 , numeric_expression2 for  
report )
```

### Exemple

```
covariance-pop ( Cost , Margin for report)
```

Résultat

La covariance de population entre Cost et Margin.

Données du résultat

Cost	Margin	covariance-pop (Cost, Margin for report)
4	0.33	0.032384
5	0.28	0.032384
9.22	0.23	0.032384

<b>Cost</b>	<b>Margin</b>	<b>covariance-pop (Cost, Margin for report)</b>
15.93	0.28	0.032384
34.97	0.3	0.032384

### **covariance-samp**

Renvoie la covariance d'échantillon d'un ensemble de paires de nombres.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

#### **Syntaxe**

```
covariance-samp ( numeric_expression1 , numeric_expression2 )
covariance-samp ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
covariance-samp ( numeric_expression1 , numeric_expression2 for
report )
```

#### **Exemple**

```
covariance-samp ( Cost , Margin for report)
```

Résultat

La covariance d'échantillon entre Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>covariance-samp (Cost, Margin for report)</b>
4	0.33	0.04048
5	0.28	0.04048
9.22	0.23	0.04048
15.93	0.28	0.04048
34.97	0.3	0.04048

### **regression-average-x**

Renvoie la moyenne de la variable indépendante (numeric\_expression2) de la courbe de régression.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

#### **Syntaxe**

```
regression-average-x ( numeric_expression1 , numeric_expression2 )
regression-average-x ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
regression-average-x ( numeric_expression1 , numeric_expression2 for
report )
```

#### **Exemple**

```
regression-average-x ( Cost , Margin for report)
```

Résultat

La moyenne de Margin dans la courbe de régression pour Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-average-x (Cost, Margin for report)</b>
4	0.33	0.284
5	0.28	0.284
9.22	0.23	0.284
15.93	0.28	0.284
34.97	0.3	0.284

### **regression-average-y**

Renvoie la moyenne de la variable dépendante (numeric\_expression1) de la courbe de régression.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

#### **Syntaxe**

```
regression-average-y ( numeric_expression1 , numeric_expression2 )
regression-average-y ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
regression-average-y ( numeric_expression1 , numeric_expression2 for
report )
```

#### **Exemple**

```
regression-average-y ( Cost , Margin for report)
```

Résultat

La moyenne de Cost dans la courbe de régression pour Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-average-y (Cost, Margin for report)</b>
4	0.33	13.824
5	0.28	13.824
9.22	0.23	13.824
15.93	0.28	13.824
34.97	0.3	13.824

### **regression-count**

Renvoie le nombre de nombres non nuls utilisés pour tracer la courbe de régression.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

#### **Syntaxe**

```
regression-count ( numeric_expression1 , numeric_expression2 )
regression-count ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
regression-count ( numeric_expression1 , numeric_expression2 for
report )
```

#### **Exemple**

```
regression-count ( Cost , Margin for report)
```

Résultat



Le nombre de nombres non nuls utilisés pour tracer la courbe de régression pour Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-count (Cost, Margin for report)</b>
4	0.33	5
5	0.28	5
9.22	0.23	5
15.93	0.28	5
34.97	0.3	5

### **regression-intercept**

Renvoie la valeur de y-intercept de la courbe de régression. Le calcul est le suivant :  $AVG(numeric\_expression1) - REGR\_SLOPE(numeric\_expression1, numeric\_expression2) * AVG(numeric\_expression2)$

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
regression-intercept ( numeric_expression1 , numeric_expression2 )
regression-intercept ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
regression-intercept ( numeric_expression1 , numeric_expression2 for
report )
```

### **Exemple**

```
regression-intercept ( Cost , Margin for report)
```

Résultat

La valeur de y-intercept de la courbe de régression pour Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-intercept (Cost, Margin for report)</b>
4	0.33	5.18015038
5	0.28	5.18015038
9.22	0.23	5.18015038
15.93	0.28	5.18015038
34.97	0.3	5.18015038

### **regression-r2**

Renvoie le coefficient de détermination (aussi appelé "R carré" ou "qualité de l'ajustement") de la courbe de régression. Cette valeur est calculée en fonction des conditions suivantes :  $IF VAR\_POP(numeric\_expression2) = 0 THEN NULL IF VAR\_POP(numeric\_expression1) = 0 AND VAR\_POP(numeric\_expression2) <> 0 THEN 1 IF VAR\_POP(numeric\_expression1) > 0 and VAR\_POP(numeric\_expression2) <> 0 THEN POWER(CORR (numeric\_expression1, numeric\_expression2))$

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Syntaxe

```
regression-r2 ( numeric_expression1 , numeric_expression2 )  
regression-r2 ( numeric_expression1 , numeric_expression2 for  
[ all|any ] expression { , expression } )  
regression-r2 ( numeric_expression1 , numeric_expression2 for  
report )
```

## Exemple

```
regression-r2 ( Cost , Margin for report)
```

## Résultat

Le coefficient de détermination de la courbe de régression pour Cost et Margin.

## Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-r2 (Cost, Margin for report)</b>
4	0.33	0.00761514
5	0.28	0.00761514
9.22	0.23	0.00761514
15.93	0.28	0.00761514
34.97	0.3	0.00761514

## regression-slope

Renvoie la pente de la courbe de régression. Le calcul est le suivant :  
 $\text{COVAR\_POP}(\text{numeric\_expression1}, \text{numeric\_expression2}) /$   
 $\text{VAR\_POP}(\text{numeric\_expression2})$

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Syntaxe

```
regression-slope ( numeric_expression1 , numeric_expression2 )  
regression-slope ( numeric_expression1 , numeric_expression2 for  
[ all|any ] expression { , expression } )  
regression-slope ( numeric_expression1 , numeric_expression2 for  
report )
```

## Exemple

```
regression-slope ( Cost , Margin )
```

## Résultat

La pente de la courbe de régression pour Cost et Margin.

## Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-slope (Cost, Margin for report)</b>
4	0.33	30.43609023
5	0.28	30.43609023
9.22	0.23	30.43609023
15.93	0.28	30.43609023
34.97	0.3	30.43609023

## regression-sxx

Renvoie le calcul suivant après élimination des paires NULL :  
REGR\_COUNT(numeric\_expression1, numeric\_expression2) \*  
VAR\_POP(numeric\_expression2)

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
regression-sxx ( numeric_expression1 , numeric_expression2 )  
regression-sxx ( numeric_expression1 , numeric_expression2 for  
[ all|any ] expression { , expression } )  
regression-sxx ( numeric_expression1 , numeric_expression2 for  
report )
```

### Exemple

```
regression-sxx ( Cost , Margin for report)
```

### Résultat

Le calcul sxx de la courbe de régression pour Cost et Margin.

### Données du résultat

Cost	Margin	regression-sxx (Cost, Margin for report)
4	0.33	0.00532
5	0.28	0.00532
9.22	0.23	0.00532
15.93	0.28	0.00532
34.97	0.3	0.00532

## regression-sxy

Renvoie le calcul suivant après élimination des paires NULL :  
REGR\_COUNT(numeric\_expression1, numeric\_expression2) \*  
COVAR\_POP(numeric\_expression1, numeric\_expression2)

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
regression-sxy ( numeric_expression1 , numeric_expression2 )  
regression-sxy ( numeric_expression1 , numeric_expression2 for  
[ all|any ] expression { , expression } )  
regression-sxy ( numeric_expression1 , numeric_expression2 for  
report )
```

### Exemple

```
regression-sxy ( Cost , Margin for report)
```

### Résultat

Le calcul sxy de la courbe de régression pour Cost et Margin.

### Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-sxy (Cost, Margin for report)</b>
4	0.33	0.16192
5	0.28	0.16192
9.22	0.23	0.16192
15.93	0.28	0.16192
34.97	0.3	0.16192

### **regression-syy**

Renvoie le calcul suivant après élimination des paires NULL :  
 REGR\_COUNT(numeric\_expression1, numeric\_expression2) \*  
 VAR\_POP(numeric\_expression1)

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
regression-syy ( numeric_expression1 , numeric_expression2 )
regression-syy ( numeric_expression1 , numeric_expression2 for
[ all|any ] expression { , expression } )
regression-syy ( numeric_expression1 , numeric_expression2 for
report )
```

### **Exemple**

```
regression-syy ( Cost , Margin for report)
```

Résultat

Le calcul syy de la courbe de régression pour Cost et Margin.

Données du résultat

<b>Cost</b>	<b>Margin</b>	<b>regression-syy (Cost, Margin for report)</b>
4	0.33	647.15932
5	0.28	647.15932
9.22	0.23	647.15932
15.93	0.28	647.15932
34.97	0.3	647.15932

### **standard-deviation**

Renvoie l'écart type des données élémentaires sélectionnées.

### **Syntaxe**

```
standard-deviation ( expression [ auto ] )
standard-deviation ( expression for [ all|any ] expression { ,
expression } )
standard-deviation ( expression for report )
```

### **Exemple**

```
standard-deviation ( ProductCost )
```

Résultat

Renvoie une valeur indiquant l'écart type entre les coûts des produits et le coût de produit moyen.

## **standard-deviation-pop**

Calcule l'écart type de remplissage et renvoie la racine carrée de la variance de remplissage.

### **Syntaxe**

```
standard-deviation-pop ( expression [ auto ] )  
standard-deviation-pop ( expression for [ all|any ] expression { ,  
    expression } )  
standard-deviation-pop ( expression for report )
```

### **Exemple**

```
standard-deviation-pop ( ProductCost )
```

Résultat

Revoie une valeur de la racine carrée de la variance de remplissage.

## **standard-deviation-samp**

Calcule l'écart-type de l'échantillon et renvoie la racine carrée de la variance de population.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
standard-deviation-samp ( expression [ auto ] )  
standard-deviation-samp ( expression for [ all|any ] expression { ,  
    expression } )  
standard-deviation-samp ( expression for report )
```

### **Exemple**

```
standard-deviation-samp ( ProductCost )
```

Résultat

Revoie la valeur de la racine carrée de la variance de l'échantillon.

## **variance**

Revoie la variance des données élémentaires sélectionnées.

### **Syntaxe**

```
variance ( expression [ auto ] )  
variance ( expression for [ all|any ] expression { , expression } )  
variance ( expression for report )
```

### **Exemple**

```
variance ( Product Cost )
```

Résultat

Revoie une valeur indiquant l'ampleur de l'écart entre le coût réel et le coût moyen des produits.

## **variance-pop**

Revoie la variance de remplissage d'un ensemble de chiffres après la suppression des valeurs nulles dans cet ensemble.

## Syntaxe

```
variance-pop ( expression [ auto ] )  
variance-pop ( expression for [ all|any ] expression { ,  
    expression } )  
variance-pop ( expression for report )
```

## Exemple

```
variance-pop ( Qty )
```

### Résultat

Renvoie, pour chaque ligne, la variance de remplissage d'un ensemble de chiffres après la suppression des valeurs nulles dans cet ensemble.

## variance-samp

Renvoie la variance d'échantillon d'un ensemble de chiffres après suppression des valeurs nulles.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Syntaxe

```
variance-samp ( expression [ auto ] )  
variance-samp ( expression for [ all|any ] expression { ,  
    expression } )  
variance-samp ( expression for report )
```

## Exemple

```
variance-samp ( Qty )
```

### Résultat

Renvoie, pour chaque ligne, la variance d'échantillon d'un ensemble de chiffres après la suppression des valeurs nulles de cet ensemble.

## aggregate

Renvoie une valeur calculée à l'aide de la fonction d'agrégation appropriée, selon le type d'agrégation de l'expression. Cette fonction apparaît dans l'exemple de rapport Données budgétaires par rapport aux données réelles, accessible dans le volume Entrepôt de données VA (analyse).

## Syntaxe

```
aggregate ( expression [ auto ] )  
aggregate ( expression for [ all|any ] expression { , expression } )  
aggregate ( expression for report )
```

## average

Retourne la valeur moyenne des données élémentaires sélectionnées. "distinct" est une expression alternative compatible avec les versions précédentes du produit.

## Syntaxe

```
average ( [ distinct ] expression [ auto ] )  
average ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
average ( [ distinct ] expression for report )
```

## Exemple

```
average ( Sales )
```

Résultat

Renvoie la valeur moyenne de toutes les valeurs des ventes.

## count

Renvoie le nombre de données élémentaires sélectionnées, sans compter les valeurs nulles. "distinct" est une expression alternative compatible avec les versions précédentes du produit. La valeur All est prise en charge en mode DQM uniquement et permet d'éviter la présomption de double-comptage d'une donnée élémentaire d'une table de dimension.

### Syntaxe

```
count ( [ all | distinct ] expression [ auto ] )  
count ( [ all | distinct ] expression for [ all|any ] expression { ,  
  expression } )  
count ( [ all | distinct ] expression for report )
```

### Exemple

```
count ( Sales )
```

Résultat

Renvoie le nombre total d'entrées sous Ventes.

## maximum

Renvoie la valeur maximale des données élémentaires sélectionnées. "distinct" est une expression alternative compatible avec les versions précédentes du produit.

### Syntaxe

```
maximum ( [ distinct ] expression [ auto ] )  
maximum ( [ distinct ] expression for [ all|any ] expression { ,  
  expression } )  
maximum ( [ distinct ] expression for report )
```

### Exemple

```
maximum ( Sales )
```

Résultat

Renvoie la valeur maximale de toutes les valeurs des ventes.

## median

Renvoie la valeur médiane des données élémentaires sélectionnées.

### Syntaxe

```
median ( expression [ auto ] )  
median ( expression for [ all|any ] expression { , expression } )  
median ( expression for report )
```

## minimum

Renvoie la valeur minimale des données élémentaires sélectionnées. "distinct" est une expression alternative compatible avec les versions précédentes du produit.

### Syntaxe

```
minimum ( [ distinct ] expression [ auto ] )  
minimum ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
minimum ( [ distinct ] expression for report )
```

### Exemple

```
minimum ( Sales )
```

Résultat

Renvoie la valeur minimale de toutes les valeurs des ventes.

## moving-average

Renvoie une moyenne mobile par ligne pour un ensemble de valeurs donné par rapport à un nombre de lignes indiqué. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
moving-average ( numeric_expression , numeric_expression [ at  
expression { , expression } ] [ <for-option> ] [ prefilter ] )  
moving-average ( numeric_expression , numeric_expression  
[ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemple

```
moving-average ( Qty , 3 )
```

Résultat

Renvoie, pour chaque ligne, la quantité et la moyenne mobile de la ligne en cours et des deux lignes précédentes.

Données du résultat

Qty	Moving-Average (Qty, 3)
200	200
700	450
400	433.3333
200	433.3333
200	266.6667
500	300.0000

## moving-total

Renvoie une valeur totale mobile par ligne pour un ensemble de valeurs donné par rapport à un nombre de lignes indiqué. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.



## Syntaxe

```
moving-total ( numeric_expression , numeric_expression [ at  
expression { , expression } ] [ <for-option> ] [ prefilter ] )  
moving-total ( numeric_expression , numeric_expression  
[ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
moving-total ( Qty , 3 )
```

### Résultat

Renvoie, pour chaque ligne, la quantité et la valeur totale mobile de la ligne en cours et des deux lignes précédentes.

### Données du résultat

<b>Qty</b>	<b>Moving-Total (Qty, 3)</b>
200	200
700	900
400	1300
200	1300
200	800
500	900

## percentage

Renvoie le pourcentage de la valeur totale des données élémentaires sélectionnées. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles. Cette fonction apparaît dans l'exemple de rapport interactif Calcul de pourcentage (par année).

## Syntaxe

```
percentage ( numeric_expression [ at expression { , expression } ]  
[ <for-option> ] [ prefilter ] )  
percentage ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
percentage ( Sales 98 )
```

### Résultat

Renvoie le pourcentage des ventes totales de 1998 qui est attribué à chacun des commerciaux.

### Données du résultat

<b>Employee</b>	<b>Sales 98</b>	<b>Percentage</b>
Gibbons	60646	7.11%
Flertjan	62523	7.35%
Cornel	22396	2.63%

## percentile

Renvoie une valeur qui indique, selon une distribution de 1 à 100, dans quel centième se situent les données élémentaires sélectionnées. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
percentile ( numeric_expression [ at expression { , expression } ]  
[ <for-option> ] [ prefilter ] )  
percentile ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemple

```
percentile ( Sales 98 )
```

#### Résultat

Renvoie, pour chaque ligne, le pourcentage des lignes dont la valeur est égale ou inférieure à la valeur de quantité de cette ligne.

#### Données du résultat

Qty	Percentile (Qty)
800	1
700	0.875
600	0.75
500	0.625
400	0.5
400	0.5
200	0.25
200	0.25

## quantile

Renvoie le rang d'une valeur au sein de la plage que vous indiquez. Le rang est exprimé sous la forme d'entiers permettant de représenter n'importe quelle plage, par exemple de 1 (le plus élevé) à 100 (le plus bas). L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
quantile ( numeric_expression , numeric_expression [ at expression { ,  
expression } ] [ <for-option> ] [ prefilter ] )  
quantile ( numeric_expression , numeric_expression [ <for-option> ]  
[ prefilter ] )  
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemple

```
quantile ( Qty , 4 )
```

#### Résultat

Renvoie la quantité, le rang de la valeur de quantité et les valeurs de quantité séparées en 4 groupes quantiles (quartiles).

#### Données du résultat

Qty	Rank	Quantile (Qty, 4)
800	1	1
700	2	1
600	3	2
500	4	2
400	5	3
400	5	3
200	7	4
200	7	4

## quantile

Renvoie le rang d'une valeur dans un groupe de valeurs, exprimé sous forme d'entiers compris entre 1 (le plus élevé) et 4 (le plus bas). L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
quantile ( numeric_expression [ at expression { , expression } ]
[ <for-option> ] [ _prefilter ] )
quantile ( numeric_expression [ <for-option> ] [ _prefilter ] )
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemple

```
quantile ( Qty )
```

Résultat

Renvoie la quantité et le quartile de la valeur de quantité représentée sous forme d'entiers compris entre 1 (le plus élevé) et 4 (le plus bas).

Données du résultat

Qty	Quantile (Qty)
450	1
400	1
350	2
300	2
250	3
200	3
150	4
100	4

## rank

Renvoie le rang des données élémentaires sélectionnées. L'ordre de tri est facultatif ; l'ordre décroissant (DESC) est utilisé par défaut. Si deux ou plusieurs lignes ont la même valeur, cela signifie qu'il manque une valeur dans l'ordre des valeurs classées (aussi appelé classement olympique). L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles. "distinct" est une expression alternative compatible avec les versions précédentes du produit. Les valeurs nulles sont classées en dernier. Cette fonction apparaît dans l'exemple de rapport Les 10 meilleurs détaillants de 2005, accessible dans le module Entrepôt de données VA (analyse).

## Syntaxe

```
rank ( expression [ ASC|DESC ] { , expression [ ASC|DESC ] } [ at
expression { , expression } ] [ <for-option> ] [ prefilter ] )
rank ( [ distinct ] expression [ ASC|DESC ] { , expression
[ ASC|DESC ] } [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
rank ( Sales 98 )
```

### Résultat

Renvoie, pour chaque ligne, le rang de la valeur des ventes pour 1998 qui est attribuée à chacun des commerciaux. Certains nombres sont ignorés lorsque des lignes sont égales.

### Données du résultat

Employee	Sales 98	Rank
Gibbons	60000	1
Flertjan	50000	2
Cornel	50000	2
Smith	48000	4

## running-average

Renvoie la moyenne cumulative par ligne (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

## Syntaxe

```
running-average ( numeric_expression [ at expression { ,
expression } ] [ <for-option> ] [ prefilter ] )
running-average ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
running-average ( Qty )
```

### Résultat

Pour chaque ligne, renvoie la quantité et une valeur moyenne cumulative de la ligne en cours et des lignes précédentes.

### Données du résultat

Name	Qty	Avg	Running-Average for name
Smith	7	5	7
Smith	3	5	5
Smith	6	5	5.33
Smith	4	5	5
Wong	3	4	3
Wong	5	4	4

## running-count

envoie le nombre cumulatif par ligne (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
running-count ( numeric_expression [ at expression { , expression } ]  
[ <for-option> ] [ prefilter ] )  
running-count ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

### Exemple

```
running-count ( Qty )
```

#### Résultat

Renvoie, pour chaque ligne, la quantité et le nombre cumulatif de la position de la ligne en cours.

#### Données du résultat

Name	Qty	Count	Running-Count for name
Smith	7	4	1
Smith	3	4	2
Smith	6	4	3
Smith	4	4	4
Wong	3	3	1
Wong	5	3	2

## running-difference

Renvoie un écart cumulatif par ligne, c'est-à-dire la différence entre la valeur de la ligne en cours et celle de la ligne qui la précède (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
running-difference ( numeric_expression [ at expression { ,  
expression } ] [ <for-option> ] [ prefilter ] )  
running-difference ( numeric_expression [ <for-option> ]  
[ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

### Exemple

```
running-difference ( Qty )
```

#### Résultat

Renvoie, pour chaque ligne, la quantité et un écart cumulatif entre la valeur de la ligne en cours et de la ligne précédente.

#### Données du résultat

Name	Qty	Running-Difference for name
Smith	7	NULL
Smith	3	-4
Smith	6	3
Smith	4	-2
Wong	3	-1
Wong	5	2

## running-maximum

Renvoie la valeur maximale cumulative par ligne (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
running-maximum ( numeric_expression [ at expression { ,
  expression } ] [ <for-option> ] [ prefilter ] )
running-maximum ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

### Exemple

```
running-maximum ( Qty )
```

### Résultat

Renvoie, pour chaque ligne, la quantité et une valeur maximale cumulative de la ligne en cours et des lignes précédentes.

### Données du résultat

Name	Qty	Max	Running-Maximum (Qty) for name
Smith	2	7	2
Smith	3	7	3
Smith	6	7	6
Smith	7	7	7
Wong	3	5	3
Wong	5	5	5

## running-minimum

Renvoie la valeur minimale cumulative par ligne (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
running-minimum ( numeric_expression [ at expression { ,
  expression } ] [ <for-option> ] [ prefilter ] )
running-minimum ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
running-minimum ( Qty )
```

Résultat

Renvoie, pour chaque ligne, la quantité et une valeur minimale cumulative de la ligne en cours et des lignes précédentes.

Données du résultat

Name	Qty	Min	Running-Minimum (Qty) for name
Smith	7	2	7
Smith	3	2	3
Smith	6	2	3
Smith	2	2	2
Wong	4	3	4
Wong	5	3	4

## running-total

Renvoie un total cumulatif par ligne (y compris la ligne en cours) pour un ensemble de valeurs. L'option "<for-option>" définit la portée de la fonction. L'option "at" définit le niveau d'agrégation et peut être utilisée seulement dans le contexte de sources de données relationnelles.

### Syntaxe

```
running-total ( numeric_expression [ at expression { ,  
  expression } ] [ <for-option> ] [ prefilter ] )  
running-total ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

## Exemple

```
running-total ( Qty )
```

Résultat

Renvoie, pour chaque ligne, la quantité et une valeur totale cumulative de la ligne en cours et des lignes précédentes.

Données du résultat

Name	Qty	Total	Running-Total (Qty) for name
Smith	2	18	2
Smith	3	18	5
Smith	6	18	11
Smith	7	18	18
Wong	3	12	3
Wong	5	12	8

## total

Renvoie la valeur totale des données élémentaires sélectionnées. "distinct" est une expression alternative compatible avec les versions précédentes du produit. Cette

fonction apparaît dans l'exemple de rapport Données budgétaires par rapport aux données réelles, accessible dans le volume Entrepôt de données VA (analyse).

### Syntaxe

```
total ( [ distinct ] expression [ auto ] )  
total ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
total ( [ distinct ] expression for report )
```

### Exemple

```
total ( Sales )
```

Résultat

Renvoie la valeur totale de toutes les valeurs des ventes.

---

## Récapitulatifs des membres

Cette liste contient des fonctions prédéfinies qui renvoient une valeur récapitulative unique pour un ensemble de membres ou une valeur récapitulative distincte pour chaque membre d'un ensemble de membres.

### aggregate

Renvoie une valeur calculée à l'aide de la fonction d'agrégation appropriée, selon le type d'agrégation de l'expression.

#### Syntaxe

```
aggregate ( < currentMeasure|numeric_expression > within set  
set_expression )  
aggregate ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

### average

Retourne la valeur moyenne des données élémentaires sélectionnées.

#### Syntaxe

```
average ( < currentMeasure|numeric_expression > within set  
set_expression )  
average ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

#### Exemple

```
average ( Sales )
```

Résultat

Renvoie la valeur moyenne de toutes les valeurs des ventes.

### count

Renvoie le nombre de données élémentaires sélectionnées, sans compter les valeurs nulles.



## Syntaxe

```
count ( < currentMeasure|numeric_expression > within set  
set_expression )  
count ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## Exemple

```
count ( Sales )
```

Résultat

Renvoie le nombre total d'entrées sous Ventes.

## maximum

Renvoie la valeur maximale des données élémentaires sélectionnées.

## Syntaxe

```
maximum ( < currentMeasure|numeric_expression > within set  
set_expression )  
maximum ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## Exemple

```
maximum ( Sales )
```

Résultat

Renvoie la valeur maximale de toutes les valeurs des ventes.

## median

Renvoie la valeur médiane des données élémentaires sélectionnées.

## Syntaxe

```
median ( < currentMeasure|numeric_expression > within set  
set_expression )  
median ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## minimum

Renvoie la valeur minimale des données élémentaires sélectionnées.

## Syntaxe

```
minimum ( < currentMeasure|numeric_expression > within set  
set_expression )  
minimum ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## Exemple

```
minimum ( Sales )
```

Résultat

Renvoie la valeur minimale de toutes les valeurs des ventes.

## percentage

La valeur renvoyée représente le pourcentage de la valeur totale des données élémentaires sélectionnées.

### Syntaxe

```
percentage ( numeric_expression [ tuple member_expression { ,  
member_expression } ] within set set_expression )
```

### Exemple

```
percentage ( [gosales].[sales measures].[quantity] tuple [gosales].  
[Staff].[].[department] -> [West] within set children ( [gosales].  
[Staff].[].[Staff] ) )
```

## percentile

Renvoie une valeur qui indique, selon une distribution de 0 à 100, dans quel centième se situent les données élémentaires sélectionnées.

### Syntaxe

```
percentile ( numeric_expression [ tuple member_expression { ,  
member_expression } ] within set set_expression )
```

## quantile

Renvoie le rang d'une valeur pour la plage indiquée. Le rang est exprimé sous la forme d'entiers permettant de représenter n'importe quelle plage, par exemple de 1 (le plus élevé) à 100 (le plus bas).

### Syntaxe

```
quantile ( numeric_expression , numeric_expression [ tuple  
member_expression { , member_expression } ] within set  
set_expression )
```

## quartile

Renvoie le rang d'une valeur dans un groupe de valeurs, exprimé sous forme d'entiers compris entre 1 (le plus élevé) et 4 (le plus bas).

### Syntaxe

```
quartile ( numeric_expression [ tuple member_expression { ,  
member_expression } ] within set set_expression )
```

## rank

Renvoie le rang des données élémentaires sélectionnées. Le type de classement renvoyé (olympique, dense ou séquentiel) dépend de la source de données. L'ordre de tri est facultatif ; DESC est utilisé par défaut.

### Syntaxe

```
rank ( numeric_expression [ ASC|DESC ] [ tuple member_expression { ,  
member_expression } ] within set set_expression )
```

### Exemple

```
rank ( [gosales].[sales measures].[quantity] tuple [gosales].[Staff].  
[].[department] -> [West] within set children ( [gosales].[Staff].[].  
[Staff] ) )
```

## standard-deviation

Renvoie l'écart type des données élémentaires sélectionnées.

### Syntaxe

```
standard-deviation ( < currentMeasure|numeric_expression > within set  
set_expression )  
standard-deviation ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## standard-deviation-pop

Renvoie la valeur représentant la population d'écart type des données élémentaires sélectionnées.

### Syntaxe

```
standard-deviation-pop ( < currentMeasure|numeric_expression > within  
set set_expression )  
standard-deviation-pop ( < currentMeasure|numeric_expression > within  
< detail|aggregate > expression )
```

## total

Renvoie la valeur totale des données élémentaires sélectionnées.

### Syntaxe

```
total ( < currentMeasure|numeric_expression > within set  
set_expression )  
total ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## variance

Renvoie la variance des données élémentaires sélectionnées.

### Syntaxe

```
variance ( < currentMeasure|numeric_expression > within set  
set_expression )  
variance ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

## variance-pop

La valeur renvoyée représente la population de variance entre les données élémentaires sélectionnées.

### Syntaxe

```
variance-pop ( < currentMeasure|numeric_expression > within set  
set_expression )  
variance-pop ( < currentMeasure|numeric_expression > within <  
detail|aggregate > expression )
```

---

## Constantes

Une constante est une valeur fixe que vous pouvez utiliser dans une expression.

### date

Insère la date en cours du système.

## **date-time**

Insère la date et l'heure en cours du système.

## **Heure avec fuseau horaire (time with time zone)**

Insère une heure égale à zéro avec un fuseau horaire.

## **Horodatage avec fuseau horaire (timestamp with time zone)**

Insère un exemple d'horodatage avec fuseau horaire.

## **interval**

Insère un intervalle à zéro : 000 00:00:00.000.

## **Intervalle de type Année (interval year)**

Insère un intervalle de type Année égal à zéro : 0 year.

## **Intervalle de type Mois (interval month)**

Insère un intervalle de type Mois égal à zéro : 0 month.

## **Intervalle de type Année à mois (interval year to month)**

Insère un intervalle de type Année à mois égal à zéro : 0000-00 year to month.

## **Intervalle de type Jour (interval day)**

Insère un intervalle de type Jour égal à zéro : 0 day.

## **Intervalle de type Heure (interval hour)**

Insère un intervalle de type Heure égal à zéro : 0 hour.

## **Intervalle de type Minute (interval minute)**

Insère un intervalle de type Minute égal à zéro : 0 minute.

## **Intervalle de type Seconde (interval second)**

Insère un intervalle de type Seconde égal à zéro : 0 second.

## **Intervalle de type Jour à heure (interval day to hour)**

Insère un intervalle de type Jour à heure égal à zéro : 0 00 day to hour.

## **Intervalle de type Jour à minute (interval day to minute)**

Insère un intervalle de type Jour à minute égal à zéro : 0 00:00 day to minute.

## **Intervalle de type Jour à seconde (interval day to second)**

Insère un intervalle de type Jour à seconde égal à zéro : 0 00:00:00.000000000 day to second.

## **Intervalle de type Heure à minute (interval hour to minute)**

Insère un intervalle de type Heure à minute égal à zéro : 00:00 hour to minute.

## Intervalle de type Heure à seconde (interval hour to second)

Insère un intervalle de type Heure à seconde égal à zéro : 00:00:00.000000000 hour to second.

## Intervalle de type minute à seconde (interval minute to second)

Insère un intervalle de type Minute à seconde égal à zéro : 00:00.000000000 minute to second.

## null

Insère une valeur nulle si les conditions d'expression ne sont pas satisfaites.

## number

Insère le nombre 0, qui peut ensuite être remplacé par une valeur numérique.

## string

Insère une chaîne vide sous forme de deux guillemets entre lesquels vous pouvez saisir une chaîne.

## time

Insère l'heure en cours du système.

---

## Syntaxes

Cette liste contient des syntaxes et des modèles que vous pouvez utiliser pour créer une expression. Les modèles combinent plusieurs fonctions au sein d'un groupe. Le modèle CASE (recherche), par exemple, inclut les fonctions Case, When, Else et End.

### if then else

Cette syntaxe correspond au modèle d'une instruction IF.THEN...ELSE. Cette syntaxe apparaît dans l'exemple de rapport Les 10 meilleurs détaillants de 2005, accessible dans le module Entrepôt de données VA (analyse).

#### Syntaxe

```
IF ([Country] = 'Canada') THEN ([List Price] * 0.60) ELSE ([List Price])
```

### in\_range

Modèle d'une expression in\_range.

#### Syntaxe

```
[code] IN_RANGE { :30 , 40, 50, 999: }
```

#### Exemple 1

```
[code] IN_RANGE { 5 }
```

Résultat

Equivalent à [code] = 5.

### Exemple 2

```
[code] IN_RANGE { 5: }
```

Résultat

Equivalent à [code] >= 5.

### Exemple 3

```
[code] IN_RANGE { :5 }
```

Résultat

Equivalent à [code] <= 5.

### Exemple 4

```
[code] IN_RANGE { 5:10 }
```

Résultat

Equivalent à ( [code] >= 5 et [code] <= 10 ).

### Exemple 5

```
[code] IN_RANGE { :5,10,20: }
```

Résultat

Equivalent à ( [code] <= 5 ou [code] = 10 ou [code] >= 20 ).

## CASE (recherche)

Cette syntaxe correspond au modèle d'une instruction CASE recherchée comprenant les fonctions CASE, WHEN, ELSE et END.

### Syntaxe

```
CASE WHEN [Country] = 'Canada' THEN ([List Price] * 0.60) WHEN  
[CountryCode] > 100 THEN [List Price] * 0.80  
ELSE [List Price] END
```

## CASE (simple)

Cette syntaxe correspond au modèle d'une instruction CASE simple comprenant les fonctions CASE, WHEN, ELSE et END.

### Syntaxe

```
CASE [Country] WHEN 'Canada' THEN ([List Price] * 0.60) WHEN  
'Australia' THEN [List Price] * 0.80  
ELSE [List Price] END
```

---

## Fonctions de date-heure selon le principe des jours ouvrables

Cette liste contient des fonctions métier permettant d'exécuter des calculs de date et d'heure.

### \_add\_seconds

Renvoie l'heure ou l'horodatage, en fonction du format de "expression\_heure", qui résulte de l'ajout de "expression\_entier" secondes à "expression\_heure".

## Syntaxe

```
_add_seconds ( time_expression, integer_expression )
```

### Exemple 1

```
_add_seconds ( 13:04:59 , 1 )
```

Résultat

```
13:05:00
```

### Exemple 2

```
_add_seconds ( 2002-04-30 12:10:10.000, 1 )
```

Résultat

```
2002-04-30 12:10:11.000
```

### Exemple 3

```
_add_seconds ( 2002-04-30 00:00:00.000, 1/100 )
```

Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Résultat

```
2002-04-30 00:00:00.010
```

## \_add\_minutes

Renvoie l'heure ou l'horodatage, en fonction du format de "expression\_heure", qui résulte de l'ajout de "expression\_entier" minutes à "expression\_heure".

## Syntaxe

```
_add_minutes ( time_expression, integer_expression )
```

### Exemple 1

```
_add_minutes ( 13:59:00 , 1 )
```

Résultat

```
14:00:00
```

### Exemple 2

```
_add_minutes ( 2002-04-30 12:59:10.000, 1 )
```

Résultat

```
2002-04-30 13:00:10.000
```

### Exemple 3

```
_add_minutes ( 2002-04-30 00:00:00.000, 1/60 )
```

Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Résultat

2002-04-30 00:00:01.000

## **\_add\_hours**

Renvoie l'heure ou l'horodatage, en fonction du format de "expression\_heure", qui résulte de l'ajout de "expression\_entier" heures à "expression\_heure".

### **Syntaxe**

```
_add_hours ( time_expression, integer_expression )
```

### **Exemple 1**

```
_add_hours ( 13:59:00 , 1 )
```

Résultat

14:59:00

### **Exemple 2**

```
_add_hours ( 2002-04-30 12:10:10.000, 1 )
```

Résultat

2002-04-30 13:10:10.000,

### **Exemple 3**

```
_add_hours ( 2002-04-30 00:00:00.000, 1/60 )
```

Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Résultat

2002-04-30 00:01:00.000

## **\_add\_days**

Renvoie la date ou l'horodatage, en fonction du format de "expression\_date", lequel résulte de l'ajout de "expression\_entier" jours à "expression\_date".

### **Syntaxe**

```
_add_days ( date_expression, integer_expression )
```

### **Exemple 1**

```
_add_days ( 2002-04-30 , 1 )
```

Résultat

2002-05-01

### **Exemple 2**

```
_add_days ( 2002-04-30 12:10:10.000, 1 )
```

Résultat

2002-05-01 12:10:10.000



### Exemple 3

```
_add_days ( 2002-04-30 00:00:00.000, 1/24 )
```

Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Résultat

```
2002-04-30 01:00:00.000
```

## \_add\_months

Ajoute "expression\_entier" mois à "expression\_date". Si le mois obtenu comporte moins de jours que le composant Jour du mois, le dernier jours du mois obtenu est renvoyé. Dans tous les autres cas, la valeur renvoyée comporte le même composant Jour du mois que "expression\_date".

### Syntaxe

```
_add_months ( date_expression, integer_expression )
```

### Exemple 1

```
_add_months ( 2012-04-15 , 3 )
```

Résultat

```
2012-07-15
```

### Exemple 2

```
_add_months ( 2012-02-29 , 1 )
```

Résultat

```
2012-03-29
```

### Exemple 3

```
_last_of_month ( _add_months ( 2012-02-29 , 1 ) )
```

Résultat

```
2012-03-31
```

### Exemple 4

```
_add_months ( 2012-01-31 , 1 )
```

Résultat

```
2012-02-29
```

### Exemple 5

```
_add_months ( 2002-04-30 12:10:10.000 , 1 )
```

Résultat

```
2002-05-30 12:10:10.000
```

## **\_add\_years**

Ajoute "expression\_entier" années à "expression\_date". Si "expression\_date" correspond au 29 février et que l'année obtenue n'est pas une année bissextile, la date du 28 février est affectée au jour obtenu. Dans tous les autres cas, la valeur renvoyée comporte le même jour et le même mois que "expression\_date".

### **Syntaxe**

```
_add_years ( date_expression, integer_expression )
```

### **Exemple 1**

```
_add_years ( 2012-04-15 , 1 )
```

Résultat

2013-04-15

### **Exemple 2**

```
_add_years ( 2012-02-29 , 1 )
```

Résultat

2013-02-28

### **Exemple 3**

```
_add_years ( 2002-04-30 12:10:10.000 , 1 )
```

Résultat

2003-04-30 12:10:10.000

## **\_age**

Renvoie un nombre qui est obtenu en soustrayant "expression\_date" de la date du jour. La valeur est renvoyée sous la forme AAAAMMMJJ, où AAAA, MM et JJ représentent respectivement le nombre d'années, le nombre de mois et le nombre de jours.

### **Syntaxe**

```
_age ( date_expression )
```

### **Exemple**

```
_age ( 1990-04-30 ) (if today's date is 2003-02-05)
```

Résultat

120906, c'est-à-dire 12 ans, 9 mois et 6 jours.

## **\_day\_of\_week**

Renvoie le jour de la semaine (chiffre compris entre 1 et 7). Le premier jour de la semaine (jour 1) dépend de la valeur saisie dans le deuxième paramètre (chiffre compris entre 1 et 7), où 1 correspond au lundi et 7 correspond au dimanche. Notez que dans la norme ISO 8601, une semaine commence le lundi (le jour 1).

### **Syntaxe**

`_day_of_week ( date_expression, integer )`

### **Exemple**

`_day_of_week ( 2003-01-01 , 1 )`

Résultat

3

## **\_day\_of\_year**

Renvoie le jour de l'année (1 à 366) contenu dans "expression\_date". Egalement appelé jour julien.

### **Syntaxe**

`_day_of_year ( date_expression )`

### **Exemple**

`_day_of_year ( 2003-03-01 )`

Résultat

61

## **\_days\_between**

Renvoie un nombre positif ou négatif représentant le nombre de jours entre "expression\_date1" et "expression\_date2". Si "expression\_date1" est antérieure à "expression\_date2", un nombre négatif est obtenu.

### **Syntaxe**

`_days_between ( date_expression1 , date_expression2 )`

### **Exemple**

`_days_between ( 2002-04-30 , 2002-06-21 )`

Résultat

-52

## **\_days\_to\_end\_of\_month**

Renvoie un nombre représentant le nombre de jours restant dans le mois représenté par "expression\_date".

### **Syntaxe**

`_days_to_end_of_month ( date_expression )`

### **Exemple**

`_days_to_end_of_month ( 2002-04-20 14:30:22.123 )`

Résultat

10

## **\_first\_of\_month**

Renvoie une date ou un horodatage, en fonction de l'argument, en convertissant "expression\_date" en une date avec la même année et le même mois, mais en affectant la valeur 1 au jour.

### **Syntaxe**

```
_first_of_month ( date_expression )
```

### **Exemple 1**

```
_first_of_month ( 2002-04-20 )
```

Résultat

2002-04-01

### **Exemple 2**

```
_first_of_month ( 2002-04-20 12:10:10.000 )
```

Résultat

2002-04-01 12:10:10.000

## **\_last\_of\_month**

Renvoie une date ou un horodatage, en fonction de l'argument, correspondant au dernier jour du mois représenté par "expression\_date".

### **Syntaxe**

```
_last_of_month ( date_expression )
```

### **Exemple 1**

```
_last_of_month ( 2002-01-14 )
```

Résultat

2002-01-31

### **Exemple 2**

```
_last_of_month ( 2002-01-14 12:10:10.000 )
```

Résultat

2002-01-31 12:10:10.000

## **\_make\_timestamp**

Renvoie un horodatage construit à partir des expressions "expression\_entier1" (année), "expression\_entier2" (mois) et "expression\_entier3" (jour). La valeur par défaut de la portion heure est 00:00:00.000.

### **Syntaxe**

```
_make_timestamp ( integer_expression1, integer_expression2,  
integer_expression3 )
```

## Exemple

```
_make_timestamp ( 2002 , 01 , 14 )
```

Résultat

```
2002-01-14 00:00:00.000
```

## \_months\_between

Renvoie un nombre entier positif ou négatif représentant le nombre de mois entre "expression\_date1" et "expression\_date2". Si "expression\_date1" est antérieure à "expression\_date2", un nombre négatif est obtenu.

## Syntaxe

```
_months_between ( date_expression1, date_expression2 )
```

## Exemple

```
_months_between ( 2002-04-03 , 2002-01-30 )
```

Résultat

```
2
```

## \_shift\_timezone

Convertit une valeur d'horodatage d'un fuseau horaire vers un autre fuseau horaire. Cette fonction prend en compte l'heure d'été, le cas échéant. Si le premier argument est de type "horodatage", le deuxième et le troisième arguments représentent les fuseaux horaires "source" et "cible", respectivement. Si le premier argument est de type "horodatage avec un fuseau horaire", le fuseau horaire "source" est déjà impliqué dans le premier argument avant que le deuxième argument ne représente le fuseau horaire "cible". Le type de données du premier argument détermine également le type de données de la valeur de retour. Le deuxième et le troisième arguments sont de type "chaîne" et représentent des identificateurs de fuseau horaire. Vous trouverez ci-dessous une liste répertoriant ces identificateurs. Remarque : l'utilisation de cette fonction entraîne un traitement local.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Syntaxe

```
_shift_timezone ( timestamp_value , from_time_zone ,  
target_time_zone )  
_shift_timezone ( timestamp_with_time_zone_value , target_time_zone )
```

## Exemple 1

```
_shift_timezone( 2013-06-30 12:00:00 , 'EST' , 'GMT' )
```

Résultat

```
2013-06-30 16:00:00
```

## Exemple 2

```
_shift_timezone( 2013-11-30 12:00:00-05:00 , 'PST' )
```

Résultat

2013-11-30 09:00:00-08:00

### Exemple 3

Time zone abbreviations:

#### Données du résultat

GMT (GMT+00:00) Greenwich Mean Time  
UTC (GMT+00:00) Coordinated Universal Time  
WET (GMT+00:00) Western Europe Time: Lisbon, Faeroe Islands, Canary Islands  
ECT (GMT+01:00) European Central Time: Amsterdam, Brussels, Paris, Rome, Vienna  
MET (GMT+01:00) Middle European Time  
ART (GMT+02:00) Egypt Time: Cairo, Damascus, Beirut, Amman, Nicosia  
CAT (GMT+02:00) Central African Time: Johannesburg, Blantyre, Harare, Tripoli  
EET (GMT+02:00) Eastern Europe Time: Athens, Kiev, Sofia, Minsk, Bucharest, Vilnius, Tallinn  
EAT (GMT+03:00) East Africa Time: Addis Ababa, Asmera, Kampala, Nairobi, Mogadishu, Khartoum  
NET (GMT+04:00) Near East Time  
PLT (GMT+05:00) Pakistan Lahore Time  
IST (GMT+05:30) Indian Time  
BST (GMT+06:00) Bangladesh Time  
VST (GMT+07:00) Vietnam Time  
CTT (GMT+08:00) Asia, Hong Kong S.A.R. of China  
JST (GMT+09:00) Japan Time: Tokyo  
ACT (GMT+09:30) Australian Central Time: Darwin  
AET (GMT+10:00) Australian Eastern Time: Sydney, Melbourne, Canberra  
SST (GMT+11:00) Solomon Time  
AGT (GMT-03:00) Argentina Time  
BET (GMT-03:00) Brazil Eastern Time: Sao Paulo, Buenos Aires  
CNT (GMT-03:30) Newfoundland Time: St. Johns  
PRT (GMT-04:00) Puerto Rico and U.S. Virgin Islands Time  
EST (GMT-05:00) Eastern Time: Ottawa, New York, Toronto, Montreal, Jamaica, Porto Acre  
CST (GMT-06:00) Central Time: Chicago, Cambridge Bay, Mexico City  
MST (GMT-07:00) Mountain Time: Edmonton, Yellowknife, Chihuahua  
PST (GMT-08:00) Pacific Time: Los Angeles, Tijuana, Vancouver  
AST (GMT-09:00) Alaska Time: Anchorage, Juneau, Nome, Yakutat  
HST (GMT-10:00) Hawaii Time: Honolulu, Tahiti  
MIT (GMT-11:00) Midway Islands Time: Midway, Apia, Niue, Pago Pago

### Exemple 4

A customized time zone identifier may also be used, using the format GMT(+|-)HH:MM. For example, GMT-06:30 or GMT+02:00.

A more complete

list of time zone identifiers (including longer form identifiers such as "Europe/Amsterdam") may be found in the "i18n\_res.xml" file from the product's configuration folder.

## \_week\_of\_year

Renvoie le numéro de semaine de l'année contenu dans "expression\_date" en fonction de la norme ISO 8601. La semaine 1 correspond à la première semaine de l'année qui compte un jeudi, ce qui correspond à la semaine du 4 janvier. Une semaine commence le lundi (jour 1) et se termine le dimanche (jour 7).

### Syntaxe

\_week\_of\_year ( date\_expression )

### Exemple

```
_week_of_year ( 2003-01-01 )
```

Résultat

1

### **`_years_between`**

Renvoie un nombre entier positif ou négatif représentant le nombre d'années entre "expression\_date1" et "expression\_date2". Si "expression\_date1" est antérieure à "expression\_date2", un nombre négatif est obtenu.

### Syntaxe

```
_years_between ( date_expression1, date_expression2 )
```

### Exemple

```
_years_between ( 2003-01-30 , 2001-04-03 )
```

Résultat

1

### **`_ymdint_between`**

Renvoie un nombre représentant la différence entre "expression\_date1" et "expression\_date2". La valeur est renvoyée sous la forme AAAAMMJJ, où AAAA, MM et JJ représentent respectivement le nombre d'années, le nombre de mois et le nombre de jours.

### Syntaxe

```
_ymdint_between ( date_expression1 , date_expression2 )
```

### Exemple

```
_ymdint_between ( 1990-04-30 , 2003-02-05 )
```

Résultat

120906, c'est-à-dire 12 ans, 9 mois et 6 jours.

---

## Fonctions de bloc

Cette liste contient des fonctions utilisées pour accéder aux membres d'un ensemble, habituellement dans le contexte d'Analysis Studio.

### **`_firstFromSet`**

Renvoie les premiers membres trouvés dans la définition de "numeric\_expression\_maximum" + "numeric\_expression\_overflow". Si le résultat excède "numeric\_expression\_maximum" + "numeric\_expression\_overflow", seul le nombre maximal de membres est renvoyé. Dans le cas d'un ensemble comportant seulement quelques membres de plus que la valeur numeric\_expression\_maximum spécifiée, la valeur numeric\_expression\_overflow permet d'inclure le petit ensemble de membres supplémentaires. Si l'ensemble comporte davantage de membres que ce qui est autorisé par la valeur numeric\_expression\_overflow, seul le nombre de membres déterminé par la valeur numeric\_expression\_maximum est renvoyé.

## Syntaxe

```
_firstFromSet ( set_expression , numeric_expression_maximum ,  
numeric_expression_overflow )
```

### Exemple 1

```
_firstFromSet ( [great_outdoors_company].[Products].[Products].  
[Product line] , 2 , 8 )
```

Résultat

Renvoie les cinq membres de l'ensemble Ligne de produits. Les deux premiers membres sont renvoyés parmi les valeurs maximales et les trois membres suivants sont renvoyés en tant que valeurs de dépassement.

Données du résultat

```
Camping Equipment  
Golf Equipment  
Mountaineering Equipment  
Outdoor Protection  
Personal Accessories
```

### Exemple 2

```
_firstFromSet ( [great_outdoors_company].[Products].[Products].  
[Product line] , 2 , 2 )
```

Résultat

Matériel de camping, matériel de golf

## \_remainderSet

Renvoie l'ensemble contenant "expression\_membre" lorsque la taille de "expression\_ensemble" est supérieure à la valeur "expression\_numérique" ; cela signifie qu'un nouveau membre est généré si le nombre de membres dans "expression\_ensemble" est supérieur à la valeur "expression\_numérique" définie.

## Syntaxe

```
_remainderSet ( member_expression, set_expression ,  
numeric_expression )
```

### Exemple

```
_remainderSet ( member ( aggregate ( currentMeasure within set  
[great_outdoors_company].[Products].[Products].[Product line] ) ,  
'Product Aggregate' , 'Product Aggregate' , [great_outdoors_company].  
[Products].[Products] ) , [great_outdoors_company].[Products].  
[Products].[Product line] , 1 )
```

Résultat

Quantité vendue pour Agrégat des produits.



### Chaîne d'Aster DATA

#### **overlay**

Renvoie "string\_exp1" en remplaçant "string\_exp2" avec la position de caractère numeric\_exp.

#### **Syntaxe**

```
overlay ( string_exp1, string_exp2, numeric_exp1 [, numeric_exp2] )
```

#### **btrim**

Renvoie string\_exp1 après avoir supprimé la chaîne de caractères la plus longue dans string\_exp2.

#### **Syntaxe**

```
btrim( string_exp1 [, string_exp2] )
```

#### **initcap**

Renvoie "string\_exp" avec la mise en forme suivante : la première lettre de chaque mot est en majuscule et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

#### **Syntaxe**

```
initcap ( string_exp )
```

#### **lpad**

Renvoie "string\_exp1" complétée à droite par "integer\_exp" occurrences de "string\_exp2". Si la longueur de "string\_exp1" dépasse "integer\_exp", la partie appropriée de "string\_exp1" est renvoyée.

#### **Syntaxe**

```
lpad ( string_exp1, integer_exp [ , string_exp2 ] )
```

#### **ltrim**

Renvoie la chaîne "string\_exp1" dans laquelle les premiers caractères sont supprimés jusqu'au premier caractère qui n'appartient pas à "string\_exp2", par exemple, ltrim('xyxXxyAB', 'xy') renvoie 'XxyAB'.

#### **Syntaxe**

```
ltrim ( string_exp1 [ , string_exp2 ] )
```

#### **md5**

Renvoie le hachage MD5 de "string\_exp1".

#### **Syntaxe**

```
md5 ( string_exp1 )
```

#### **to\_hex**

Renvoie une représentation sous forme de chaîne hexadécimale de "numeric\_exp1".

#### **Syntaxe**

```
to_hex ( numeric_exp1 )
```

### **repeat**

Renvoie la chaîne "string\_exp" répétée le nombre de fois déterminé par "numeric\_exp1".

### **Syntaxe**

```
repeat ( string_exp, numeric_exp1 )
```

### **replace**

Renvoie "string\_exp" après avoir remplacé "string\_exp2" par "string\_exp3".

### **Syntaxe**

```
replace( string_exp, string_exp2, string_exp3)
```

### **rpad**

Renvoie "string\_exp1" complétée à droite par "integer\_exp" occurrences de "string\_exp2". Si la longueur de "string\_exp1" dépasse "integer\_exp", la partie appropriée de "string\_exp1" est renvoyée. Si "string\_exp2" n'est pas spécifiée, des espaces sont utilisés.

### **Syntaxe**

```
rpad ( string_exp1, integer_exp [ , string_exp2 ] )
```

### **rtrim**

Renvoie la chaîne "string\_exp1" dans laquelle les caractères de fin ont été supprimés après le dernier caractère qui ne figure pas dans "string\_exp2", par exemple, rtrim('ABxXxyx', 'xy') renvoie 'ABxX'. Si "string\_exp2" n'est pas spécifiée, les caractères espace de fin sont supprimés.

### **Syntaxe**

```
rtrim ( string_exp1 [ , string_exp2 ] )
```

### **split\_part**

Renvoie la zone "numeric\_exp" en séparant "string\_exp1" dans "string\_exp2".

### **Syntaxe**

```
split_part ( string_exp1 , string_exp2 , numeric_exp )
```

## **Formatage du type de données Aster DATA**

### **to\_char**

Renvoie "exp" sous forme de chaîne de type "string\_exp". "Exp" peut être une valeur de date ou une valeur numérique.

### **Syntaxe**

```
to_char ( exp , string_exp )
```

### **to\_date**

Convertit "string\_exp1" en une valeur de date, comme indiqué par le format "string\_exp2".

### **Syntaxe**

```
to_date ( string_exp1 , string_exp2 )
```

**to\_number**

Convertit "string\_exp1" en une valeur numérique, comme indiqué par le format "string\_exp2".

**Syntaxe**

```
to_number ( string_exp1, string_exp2 )
```

**to\_timestamp**

Convertit "string\_exp1" en une valeur d'horodatage, comme indiqué par le format "string\_exp2".

**Syntaxe**

```
to_timestamp ( string_exp1, string_exp2 )
```

## Mathématiques appliquées à Aster DATA

**log**

Renvoie le logarithme en base 10 de "numeric\_exp1" ou le logarithme en base "numeric\_exp2".

**Syntaxe**

```
log ( numeric_exp1[, numeric_exp2] )
```

**ln**

Renvoie le logarithme naturel de "numeric\_exp1".

**Syntaxe**

```
ln ( numeric_exp )
```

**cbrt**

Renvoie la racine cubique de "numeric\_exp1".

**Syntaxe**

```
cbrt ( numeric_exp )
```

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

```
pi ( )
```

## Trigonométrie sur Aster DATA

**acos**

Renvoie l'arc cosinus de "numeric\_exp" en radians. L'arc cosinus est l'angle dont le cosinus est "numeric\_exp".

**Syntaxe**

```
acos ( numeric_exp )
```

**asin**

Renvoie l'arc sinus de "numeric\_exp" en radians. L'arc sinus est l'angle dont le sinus est "numeric\_exp".

**Syntaxe**

```
asin ( numeric_exp )
```

**atan**

Renvoie l'arc tangente de "numeric\_exp" en radians. L'arc tangente est l'angle dont la tangente est "numeric\_exp".

**Syntaxe**

```
atan ( numeric_exp )
```

**atan2**

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "numeric\_exp1" et "numeric\_exp2", en radians. L'arc tangente est l'angle dont la tangente est "numeric\_exp2"/"numeric\_exp1".

**Syntaxe**

```
atan2 ( numeric_exp1, numeric_exp2 )
```

**cos**

Renvoie le cosinus de "numeric\_exp", où "numeric\_exp" représente un angle exprimé en radians.

**Syntaxe**

```
cos ( numeric_exp )
```

**cot**

Renvoie la cotangente de "numeric\_exp", où "numeric\_exp" représente un angle exprimé en radians.

**Syntaxe**

```
cot ( numeric_exp )
```

**degrees**

Renvoie une valeur en degrés où "numeric\_exp" représente un angle exprimé en radians.

**Syntaxe**

```
degrees ( numeric_exp )
```

**radians**

Renvoie les radians où "numeric\_exp" correspond à un angle exprimé en degrés.

**Syntaxe**

```
radians ( numeric_exp )
```

**sin**

Renvoie le sinus de "numeric\_exp", où "numeric\_exp" correspond à un angle exprimé en radians.

**Syntaxe**

```
sin ( numeric_exp )
```

**tan**

Renvoie la tangente de "numeric\_exp", où "numeric\_exp" correspond à un angle exprimé en radians.

**Syntaxe**

```
tan ( numeric_exp )
```

**ascii**

Renvoie un nombre représentant la valeur de code ASCII du caractère le plus à gauche dans "string\_exp", par exemple, ascii('A') est 65.

**Syntaxe**

```
ascii ( string_exp )
```

**chr**

Renvoie le caractère dont la valeur de code ASCII est spécifiée par "integer\_exp". La valeur de "integer\_exp" doit être comprise entre 0 et 255.

**Syntaxe**

```
chr ( integer_exp )
```

**current\_schema**

Renvoie le nom du schéma en cours.

**Syntaxe**

```
current_schema ( )
```

**translate**

Renvoie la chaîne "string\_exp1" dans laquelle chaque occurrence de chaque caractère dans "string\_exp2" est remplacée par le caractère correspondant dans "string\_exp3".

**Syntaxe**

```
translate ( string_exp1, string_exp2, string_exp3 )
```

**date\_trunc**

Renvoie l'horodatage selon la précision spécifiée.

**Syntaxe**

```
date_trunc ( string_exp, timestamp_exp )
```

**version**

Renvoie la valeur de chaîne de la version de base de données.

**Syntaxe**

```
version ( )
```

---

**DB2****Fonctions mathématiques DB2****log**

Renvoie le logarithme naturel de "expression\_numérique".

**Syntaxe**

`log ( numeric_expression )`

**log10**

Renvoie le logarithme à base 10 de "expression\_numérique".

**Syntaxe**

`log10 ( numeric_expression )`

**rand**

Renvoie un nombre aléatoire en utilisant "expression\_entier" comme valeur de départ.

**Syntaxe**

`rand ( integer_expression )`

## Fonctions trigonométriques DB2

**acos**

Renvoie l'arc cosinus de "expression\_numérique" en radians. L'arc cosinus est l'angle dont le cosinus est "expression\_numérique".

**Syntaxe**

`acos ( numeric_expression )`

**asin**

Renvoie l'arc sinus de "expression\_numérique" en radians. L'arc sinus est l'angle dont le sinus est "expression\_numérique".

**Syntaxe**

`asin ( numeric_expression )`

**atan**

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

**Syntaxe**

`atan ( numeric_expression )`

**atanh**

Renvoie l'arc tangente hyperbolique de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`atanh ( numeric_expression )`

**atan2**

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "expression\_numérique1" et "expression\_numérique2", en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique2"/"expression\_numérique1".

**Syntaxe**

`atan2 ( numeric_expression1 , numeric_expression2 )`

**cos**

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`cos ( numeric_expression )`

**cosh**

Renvoie le cosinus hyperbolique de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`cosh ( numeric_expression )`

**cot**

Renvoie la cotangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`cot ( numeric_expression )`

**degrees**

Renvoie "expression\_numérique" radians convertis en degrés.

**Syntaxe**

`degrees ( numeric_expression )`

**sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`sin ( numeric_expression )`

**sinh**

Renvoie le sinus hyperbolique de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`sinh ( numeric_expression )`

**tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`tan ( numeric_expression )`

**tanh**

Renvoie la tangente hyperbolique de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

`tanh ( numeric_expression )`

## **ascii**

Renvoie la valeur de code ASCII du caractère le plus à gauche dans l'argument, sous la forme d'un entier.

### **Syntaxe**

```
ascii ( string_expression )
```

### **Exemple**

```
ascii ( a )
```

Résultat

Renvoie la valeur 65, qui correspond au code de la valeur ASCII "a".

## **ceiling**

Renvoie le plus petit entier supérieur ou égal à "expression\_numérique".

### **Syntaxe**

```
ceiling ( numeric_expression )
```

### **Exemple**

```
ceiling ( 0.75 )
```

Résultat

Renvoie 0.8.

## **char**

Renvoie une représentation sous forme de chaîne d'une valeur date-heure ou d'un nombre décimal.

### **Syntaxe**

```
char ( expression )
```

## **chr**

Renvoie le caractère dont la valeur de code ASCII est spécifiée par "expression\_entier". La valeur de "expression\_entier" doit être comprise entre 0 et 255.

### **Syntaxe**

```
chr ( integer_expression )
```

### **Exemple**

```
chr ( 65 )
```

Résultat

Renvoie a, le caractère correspondant au code de la valeur ASCII 65.

## **concat**

Renvoie une chaîne qui correspond à la concaténation des chaînes "expression\_chaine1" et "expression\_chaine2".



## Syntaxe

concat ( string\_expression1, string\_expression2 )

## Exemple

```
concat ( [Sales target (query)].[Sales staff].[First name], [Sales target (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie le prénom et le nom ; par exemple, Bob Smith.

## date

Renvoie une date à partir d'une valeur d'entrée unique. L'élément "expression" peut être une date représentée sous forme d'une chaîne ou d'un entier.

## Syntaxe

date ( expression )

## Exemple

```
date ( '1998-01-08' )
```

Résultat

Renvoie 8 janvier 1998.

## day

Renvoie le jour du mois (1-31) à partir de "expression\_date". L'élément "expression\_date" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

## Syntaxe

day ( date\_expression )

## Exemple

```
day ( '1998-01-08' )
```

Résultat

Renvoie 8.

## dayname

Renvoie une chaîne de caractères contenant le nom du jour propre à la source de données (par exemple, Sunday à Saturday ou Sun. à Sat. pour une source de données qui utilise l'anglais, ou Sonntag à Samstag pour une source de données qui utilise l'allemand) pour la partie jour de "expression\_date". L'élément "expression\_date" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

## Syntaxe

dayname ( date\_expression )

### **Exemple**

```
dayname ( '1998-01-08' )
```

Résultat

Renvoie Thursday.

## **dayofweek**

Renvoie le jour de la semaine contenu dans "expression\_date" sous forme d'un entier compris entre 1 et 7, où 1 représente le dimanche. L'élément "expression\_date" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

### **Syntaxe**

```
dayofweek ( date_expression )
```

### **Exemple**

```
dayofweek ( '1998-01-08' )
```

Résultat

Renvoie 5.

## **dayofweek\_iso**

Renvoie le jour de la semaine contenu dans "expression\_date" sous forme d'un entier compris entre 1 et 7, où 1 représente le lundi. L'élément "expression\_date" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

### **Syntaxe**

```
dayofweek_iso ( date_expression )
```

### **Exemple**

```
dayofweek_iso ( '1998-01-08' )
```

Résultat

Renvoie 4.

## **dayofyear**

Renvoie le jour de l'année contenu dans "expression\_date" sous forme d'un entier compris entre 1 et 366. L'élément "expression\_date" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

### **Syntaxe**

```
dayofyear ( date_expression )
```

### **Exemple**

```
dayofyear ( current_date )
```

Résultat

Renvoie le jour de l'année correspondant à la date du jour ; par exemple, si la date du jour est le 28 janvier, l'expression renvoie la valeur 28.

## days

Renvoie un entier représentant une date. L'élément "expression" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

### Syntaxe

```
days ( expression )
```

## dec

Renvoie la représentation décimale de "string-expression" en fonction des valeurs de "precision-integer" et de "scale-integer". La valeur de "decimal-character" peut être utilisée pour définir le caractère à un octet de la constante utilisée pour délimiter les nombres décimaux dans "string-expression". La valeur de "string-expression" doit être formatée en tant que constante entière ou décimale SQL. Autre syntaxe : dec ( numeric-expression [ , precision-integer [ , scale-integer ] ] ) Renvoie la représentation décimale de "string-expression" en fonction des valeurs de "precision-integer" et de "scale-integer".

### Syntaxe

```
dec ( string-expression [ , precision-integer [ , scale-integer [ ,  
decimal-character ] ] ] )
```

## decimal

Renvoie la représentation décimale de "string-expression" en fonction des valeurs de "precision-integer" et de "scale-integer". La valeur de "decimal-character" peut être utilisée pour définir le caractère à un octet de la constante utilisée pour délimiter les nombres décimaux dans "string-expression". La valeur de "string-expression" doit être formatée en tant que constante entière ou décimale SQL. Autre syntaxe : decimal ( numeric-expression [ , precision-integer [ , scale-integer ] ] ) Renvoie la représentation décimale de "string-expression" en fonction des valeurs de "precision-integer" et de "scale-integer".

### Syntaxe

```
decimal ( string-expression [ , precision-integer [ , scale-integer  
[ , decimal-character ] ] ] )
```

## difference

Renvoie une valeur entière représentant la différence entre les valeurs renvoyées par la fonction SOUNDEX spécifique de source de données pour "expression\_chaine1" et "expression\_chaine2". La valeur renvoyée se situe entre 0 et 4, où 4 représente la meilleure correspondance. Une valeur de 4 ne signifie pas que les chaînes sont égales.

### Syntaxe

```
difference ( string_expression1 , string_expression2 )
```

### Exemple 1

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales (  
query)].[Retailers].[Contact first name])
```

Résultat

0

## Exemple 2

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])
```

Résultat

4

## digits

Renvoie la représentation d'un nombre en virgule fixe sous forme de chaîne de caractères.

### Syntaxe

```
digits ( numeric_expression )
```

## double

envoie la représentation en virgule flottante d'une expression. "Expression" peut correspondre à une expression numérique ou de chaîne.

### Syntaxe

```
double ( expression )
```

## event\_mon\_state

Renvoie l'état de fonctionnement d'un moniteur d'état déterminé.

### Syntaxe

```
event_mon_state ( string_expression )
```

## float

Renvoie la représentation d'un nombre en virgule flottante.

### Syntaxe

```
float ( numeric_expression )
```

## hex

Renvoie la représentation hexadécimale d'une valeur.

### Syntaxe

```
hex ( expression )
```

## hour

Renvoie l'heure contenue dans "expression\_heure" sous la forme d'un entier compris entre 0 (minuit) et 23 (23:00). L'élément "expression\_heure" peut être une heure représentée sous forme d'une chaîne ou d'une valeur.

### Syntaxe

```
hour ( time_expression )
```

### Exemple

```
hour ( 01:22:45 )
```

Résultat

Renvoie 1.

## insert

Renvoie une chaîne résultant de la suppression de "expression2\_entier" caractères dans "expression1\_chaîne" à partir de "expression1\_entier" et de l'insertion de "expression2\_chaîne" au début de "expression1\_chaîne". Le premier caractère de la chaîne se trouve en position 1.

### Syntaxe

```
insert ( string_expression1, integer_expression1,  
integer_expression2, string_expression2 )
```

## integer

Renvoie une expression sous forme d'un entier. L'élément "expression" peut être un nombre représenté sous forme de chaîne ou de valeur numérique.

### Syntaxe

```
integer ( expression )
```

### Exemple

```
integer ( 84.95 )
```

Résultat

85

## int

Renvoie une expression sous forme d'un entier. L'élément "expression" peut être un nombre représenté sous forme de chaîne ou de valeur numérique.

### Syntaxe

```
int ( expression )
```

### Exemple

```
int ( 84.95 )
```

Résultat

85

## julian\_day

Renvoie une valeur entière représentant le nombre de jours depuis le 1 janvier, 4712 Av.J-C (le début du calendrier julien) et la valeur de date spécifiée dans "expression". L'élément "expression" peut être une date représentée sous forme d'une chaîne ou d'une valeur.

### Syntaxe

```
julian_day ( expression )
```

### Exemple

```
julian_day ( '2009-06-29' )
```

Résultat

## lcase

Renvoie "expression\_chaîne" après avoir converti tous les caractères majuscules en minuscules.

### Syntaxe

```
lcase ( string_expression )
```

### Exemple

```
lcase ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille sans majuscules.

## left

Renvoie les "expression\_entier" caractères situés le plus à gauche dans "expression\_chaîne".

### Syntaxe

```
left ( string_expression, integer_expression )
```

### Exemple

```
left ( [Sales (query)].[Sales staff].[Last name] , 3 )
```

Résultat

Renvoie les trois premiers caractères de chaque nom.

## length

Renvoie la longueur de l'opérande, exprimée en octets. Exception : les types de chaîne sur deux octets renvoient la longueur en caractères.

### Syntaxe

```
length ( expression )
```

### Exemple

```
length ( [Sales (query)].[Sales staff].[Record start date] )
```

Résultat

Renvoie la valeur 4 ; les dates renvoient toujours la valeur 4.

## locate

Renvoie la position de début de la première occurrence de "expression\_chaîne1" dans "expression\_chaîne2". La recherche débute à la position "expression\_entier" de "expression\_chaîne2". Le premier caractère dans une chaîne figure en position 1. Si "expression\_chaîne1" est introuvable, la valeur zéro est renvoyée.

### Syntaxe

```
locate ( string_expression1, string_expression2 [ ,  
integer_expression ] )
```

### **Exemple**

```
locate ( A, [Sales (query)].[Sales staff].[Last name] , 2 )
```

Résultat

Renvoie la position du caractère A dans les noms de famille, en commençant par le deuxième caractère du nom de famille.

## **long\_varchar**

Renvoie une chaîne de type LONG.

### **Syntaxe**

```
long_varchar ( string_expression )
```

## **ltrim**

Renvoie "expression\_chaine" après avoir supprimé les espaces à gauche.

### **Syntaxe**

```
ltrim ( string_expression )
```

### **Exemple**

```
ltrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à gauche.

## **microsecond**

Renvoie la partie microsecondes (unité de temps) d'une valeur. La valeur de "expression" peut être un horodatage ou une chaîne de caractères exprimant un horodatage.

### **Syntaxe**

```
microsecond ( expression )
```

### **Exemple**

```
microsecond ( 01:45:34.056 )
```

Résultat

Renvoie 056.

## **midnight\_seconds**

Renvoie une valeur entière comprise entre 0 et 86400 qui représente le nombre de secondes entre minuit et la valeur horaire indiquée dans l'argument. La valeur de "expression" peut être une valeur horaire, un horodatage ou une chaîne de caractères exprimant une heure.

### **Syntaxe**

```
midnight_seconds ( expression )
```

### **Exemple**

midnight\_seconds ( 01:45:34.056 )

Résultat

Renvoie 6334.

## **minute**

Renvoie la minute (nombre entier compris entre 0 et 59) contenue dans "expression\_heure". La valeur de "expression\_heure" peut être une heure, un horodatage ou une chaîne de caractères exprimant une heure.

### **Syntaxe**

minute ( time\_expression )

### **Exemple**

minute ( 01:45:34.056 )

Résultat

Renvoie 45.

## **month**

Renvoie le mois (entier compris entre 1 et 12) contenu dans "expression\_date".

### **Syntaxe**

month ( date\_expression )

### **Exemple**

month ( 2005-11-01 )

Résultat

Renvoie 11.

## **monthname**

Renvoie une chaîne de caractères contenant le nom du mois propre à la source de données pour la partie mois de "expression\_date" (par exemple, January à December ou Jan. à Dec. pour une source de données en anglais ou Janvier à Décembre pour une source de données en français).

### **Syntaxe**

monthname ( date\_expression )

### **Exemple**

monthname ( 2005-11-01 )

Résultat

Novembre



## quarter

Renvoie le trimestre contenu dans "expression\_date" sous forme d'un chiffre compris entre 1 et 4, où 1 représente les dates comprises entre le 1er janvier et le 31 mars.

### Syntaxe

```
quarter ( date_expression )
```

### Exemple

```
quarter ( 2005-11-01 )
```

Résultat

Renvoie 4.

## radians

Renvoie "expression\_numérique" degrés transformés en radians.

### Syntaxe

```
radians ( numeric_expression )
```

## repeat

Renvoie la chaîne "expression\_chaine" répétée le nombre de fois déterminé par "expression\_entier".

### Syntaxe

```
repeat ( string_expression, integer_expression )
```

### Exemple

```
repeat ( XYZ, 3 )
```

Résultat

Renvoie XYZXYZXYZ.

## replace

Remplace toutes les occurrences de "expression\_chaine2" par "expression\_chaine3" dans "expression\_chaine1".

### Syntaxe

```
replace ( string_expression1, string_expression2, string_expression3 )
```

### Exemple

```
replace ( [Sales (query)].[Sales staff].[Position code], A, a )
```

Résultat

Renvoie les codes de position comportant toutes les occurrences de "A" remplacées par "a".

## right

Renvoie les "expression\_entier" caractères situés le plus à droite dans "expression\_chaîne".

### Syntaxe

```
right ( string_expression, integer_expression )
```

### Exemple

```
right ( [Sales (query)].[Sales staff].[Position code], 3 )
```

Résultat

Renvoie les 3 caractères situés le plus à droite dans chaque code de position.

## round

Renvoie l'expression "expression\_numérique" arrondie à "expression\_entier" positions à droite du séparateur décimal. Si "expression\_entier" est une valeur négative, "expression\_numérique" est arrondi à la valeur absolue la plus proche avec "expression\_entier" positions à gauche du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression, integer_expression )
```

### Exemple

```
round ( 3.14159265, 3 )
```

Résultat

Renvoie 3,142.

## rtrim

Renvoie "expression\_chaîne" après avoir supprimé les espaces de fin de chaîne.

### Syntaxe

```
rtrim ( string_expression )
```

### Exemple

```
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à la fin de ces derniers.

## second

Renvoie la seconde (entier compris entre 0 et 59) contenue dans "expression\_heure".

### Syntaxe

```
second ( time_expression )
```

### **Exemple**

`second ( 01:45:34.056 )`

Résultat

Renvoie 34.

## **sign**

Renvoie un indicateur du signe de la chaîne "expression\_numérique" : +1 si "expression\_numérique" est positive, 0 si elle est nulle et -1 si elle est négative.

### **Syntaxe**

`sign ( numeric_expression )`

### **Exemple**

`sign ( [Revenue] )`

Résultat

Renvoie le signe + pour les valeurs positives et le signe - pour les valeurs négatives.

## **smallint**

Renvoie un nombre de type smallint.

### **Syntaxe**

`smallint ( expression )`

## **soundex**

Renvoie un code composé d'une chaîne de 4 caractères correspondant à une abréviation phonétique de "expression\_chaine". Cette fonction permet de déterminer si deux chaînes sont phonétiquement semblables. Par exemple, est-ce que ('SMITH') = ('SMYTH') phonétiquement parlant.

### **Syntaxe**

`soundex ( string_expression )`

## **space**

Renvoie une chaîne composée de "expression\_entier" espaces.

### **Syntaxe**

`space ( integer_expression )`

### **Exemple**

`space ( 5 )`

Résultat

Renvoie 5 espaces.

## substr

Renvoie la sous-chaîne de "expression\_chaîne" qui débute à la position "expression1\_entier" et dont la longueur est "expression2\_entier". Le premier caractère de "expression\_chaîne" se trouve en position 1.

### Syntaxe

```
substr ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

### Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## table\_name

Renvoie un nom non qualifié d'une table ou d'une vue basé sur le nom de l'objet dans "expression\_chaîne1" et le nom du schéma dans "expression\_chaîne2". Cette fonction sert à résoudre les alias.

### Syntaxe

```
table_name ( string_expression1 [ , string_expression2 ] )
```

## table\_schema

Renvoie la partie nom du schéma de la table en deux parties ou le nom de vue basé sur le nom d'objet dans "expression\_chaîne1" et le nom de schéma dans "expression\_chaîne2". Cette fonction sert à résoudre les alias.

### Syntaxe

```
table_schema ( string_expression1 [ , string_expression2 ] )
```

## time

Renvoie une heure à partir d'une valeur.

### Syntaxe

```
time ( expression )
```

## timestamp

Renvoie un horodatage à partir d'une valeur ou d'une paire de valeurs. La valeur de "expression1" doit être une date et la valeur de "expression2" doit être une heure.

### Syntaxe

```
timestamp ( expression1 [ , expression2 ] )
```

### Exemple

```
timestamp ( 11 November 2005 , 12:00:00.000000 )
```

Résultat

Renvoie 2005-11-11-12:00:00.000000.

## timestamp\_iso

Renvoie une date au format ISO (aaaa-mm-jj hh:mm:ss.nnnnnn) convertie à partir du format IBM (aaaa-mm-jj-hh.mm.ss.nnnnnn). Si "expression" correspond à une valeur d'heure, la fonction insère la valeur de la DATE DU JOUR pour les éléments de date et des zéros pour les subdivisions de l'heure.

### Syntaxe

```
timestamp_iso ( expression )
```

### Exemple

```
timestamp_iso ( 11 November 2005 , 12:00:00.000000 )
```

Résultat

Renvoie 2005-11-11 12:00:00.000000.

## timestampdiff

Renvoie un nombre approximatif d'intervalles de type "expression1" en fonction de la différence entre deux horodatages. "expression2" est le résultat de la soustraction de deux types d'horodatage et de la conversion du résultat en expression de type CHAR. Les valeurs valides pour "expression1" sont les suivantes : 1 fraction d'une seconde, 2 secondes, 4 minutes, 8 heures, 16 jours, 32 semaines, 64 mois, 128 trimestres et 256 années.

### Syntaxe

```
timestampdiff ( expression1, expression2 )
```

## to\_char

Renvoie l'horodatage sous forme de chaîne de type "expression\_chaine".

### Syntaxe

```
to_char ( timestamp_expression , string_expression )
```

## to\_date

Convertit "expression1" en une valeur date-heure de type "expression\_chaine2". Le premier argument "expression1" peut être de type chaîne ou numérique.

### Syntaxe

```
to_date ( expression1 , string_expression2 )
```

### Exemple

```
to_date ( '2013-02-27 15:13:14' , 'YYYY-MM-DD HH24:MI:SS' )
```

Résultat

Renvoie une date et une heure avec la valeur '2013-02-27 15:13:14'.

## translate

Renvoie la chaîne "expression\_chaine1" dans laquelle des caractères de "expression\_chaine3" ont été remplacés par des caractères équivalents de "expression\_chaine2". "expression\_chaine4" est un caractère unique utilisé pour

compléter la chaîne "expression\_chaine2" si celle-ci est plus courte que "expression\_chaine3". Si seule la chaîne "expression\_chaine1" est présente, cette fonction la convertit en majuscules.

### Syntaxe

```
translate ( string_expression1 [ , string_expression2,  
string_expression3 [ , string_expression4 ] ] )
```

### Exemple 1

```
translate ( 'abcdefg' )
```

Résultat

Renvoie ABCDEFG.

### Exemple 2

```
translate ( 'mnop' , n, m , - )
```

Résultat

Renvoie n-nlop.

## trunc

Renvoie l'expression "expression\_numérique1" tronquée à "expression\_numérique2" positions à droite du séparateur décimal. Si "expression\_numérique2" est négative, "expression\_numérique1" est tronquée à la valeur absolue de "expression\_numérique2" positions à gauche du séparateur décimal.

### Syntaxe

```
trunc ( numeric_expression1, numeric_expression2 )
```

### Exemple

```
trunc ( 3.14159265, 3 )
```

Résultat

Renvoie 3.141.

## truncate

Renvoie l'expression "expression\_numérique1" tronquée à "expression\_numérique2" positions à droite du séparateur décimal. Si "expression\_numérique2" est négative, "expression\_numérique1" est tronquée à la valeur absolue de "expression\_numérique2" positions à gauche du séparateur décimal.

### Syntaxe

```
truncate ( numeric_expression1, numeric_expression2 )
```

### Exemple

```
truncate ( 3141.59265, -3 )
```

Résultat

Renvoie 3.

## **ucase**

Renvoie "expression\_chaîne" après avoir converti tous les caractères minuscules en majuscules.

### **Syntaxe**

```
ucase ( string_expression )
```

### **Exemple**

```
ucase ( XY896Zbcd789 )
```

Résultat

Renvoie XY896ZBCED789.

## **value**

Renvoie le premier argument non nul (ou la valeur NULL si tous les arguments sont nuls). La fonction Value accepte au moins deux arguments.

### **Syntaxe**

```
value ( expression_list )
```

### **Exemple**

```
value ( [Unit cost], [Unit price], [Unit sale price] )
```

Résultat

Renvoie la première valeur non nulle.

## **varchar**

Renvoie "expression" sous forme de chaîne de type VARCHAR et de longueur égale à expression\_numérique.

### **Syntaxe**

```
varchar ( expression [ , numeric_expression ] )
```

## **week**

Renvoie la semaine de l'année contenue dans "expression\_date" sous forme d'un entier compris entre 1 et 53.

### **Syntaxe**

```
week ( date_expression )
```

### **Exemple**

```
week ( 11 November 2005 )
```

Résultat

Renvoie 45.

## **year**

Renvoie l'année contenue dans "expression\_date".

## Syntaxe

year ( date\_expression )

## Exemple

year ( 11 November 2005 )

Résultat

Renvoie 2005.

---

# Greenplum

## Chaîne Greenplum

### overlay

Renvoie "expression\_chaîne1" en remplaçant "expression\_chaîne2" à partir de l'emplacement du caractère "expression\_numérique".

### Syntaxe

overlay ( string\_expression1 , string\_expression2 ,  
numeric\_expression1 [ , numeric\_expression2 ] )

### btrim

Renvoie "expression\_chaîne1" après avoir supprimé la chaîne de caractères la plus longue dans "expression\_chaîne2".

### Syntaxe

btrim ( string\_expression1 [ , string\_expression2 ] )

### initcap

Renvoie "expression\_chaîne" avec la mise en forme suivante : la première lettre de chaque mot est en majuscule et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

### Syntaxe

initcap ( string\_expression )

### lpad

Renvoie "expression\_chaîne1" complétée à droite par "expression\_entier" occurrences de "expression\_chaîne2". Si la longueur de "expression\_chaîne1" dépasse "expression\_entier", la partie appropriée de "expression\_chaîne1" est renvoyée.

### Syntaxe

lpad ( string\_expression1 , integer\_expression [ , string\_expression2  
] )

### ltrim

Renvoie la chaîne "expression\_chaîne1" dans laquelle les premiers caractères sont supprimés jusqu'au premier caractère qui n'appartient pas à "expression\_chaîne2" ; par exemple, ltrim ( 'xyxXxyAB' , 'xy' ) renvoie XxyAB.

### Syntaxe

ltrim ( string\_expression1 [ , string\_expression2 ] )



## **md5**

Renvoie le hachage MD5 de "expression\_chaîne1".

### **Syntaxe**

```
md5 ( string_expression1 )
```

## **to\_hex**

Renvoie "expression\_numérique1" sous forme de chaîne hexadécimale.

### **Syntaxe**

```
to_hex ( numeric_expression1 )
```

## **repeat**

Renvoie la chaîne "expression\_chaîne" répétée le nombre de fois déterminé par "expression\_numérique1".

### **Syntaxe**

```
repeat ( string_expression , numeric_expression1 )
```

## **replace**

Renvoie "expression\_chaîne" après avoir remplacé "expression\_chaîne2" par "expression\_chaîne3".

### **Syntaxe**

```
replace ( string_expression , string_expression2 , string_expression3 )
```

## **rpad**

Renvoie "expression\_chaîne1" complétée à droite par "expression\_entier" occurrences de "expression\_chaîne2". Si la longueur de "expression\_chaîne1" dépasse "expression\_entier", la partie appropriée de "expression\_chaîne1" est renvoyée. Si "expression\_chaîne2" n'est pas spécifiée, des espaces sont utilisés.

### **Syntaxe**

```
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

## **rtrim**

Renvoie la chaîne "expression\_chaîne1" dans laquelle les caractères de fin ont été supprimés après le dernier caractère qui ne figure pas dans "expression\_chaîne2" ; par exemple, rtrim ( 'ABxXxyx' , 'xy' ) renvoie ABxX. Si "expression\_chaîne2" n'est pas spécifiée, les caractères espace de fin sont supprimés.

### **Syntaxe**

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

## **split\_part**

Renvoie la zone "expression\_numérique" en séparant "expression\_chaîne1" dans "expression\_chaîne2".

### **Syntaxe**

```
split_part ( string_expression1 , string_expression2 , numeric_expression )
```

## Formatage du type de données Greenplum

### **to\_char**

Renvoie "expression" sous forme de chaîne de type "expression\_chaîne".  
"expression" peut être une valeur de date ou une valeur numérique.

### **Syntaxe**

```
to_char ( expression , string_expression )
```

### **to\_date**

Convertit "expression\_chaîne1" en une valeur de date, comme indiqué par le format "expression\_chaîne2".

### **Syntaxe**

```
to_date ( string_expression1 , string_expression2 )
```

### **to\_number**

Convertit "expression\_chaîne1" en une valeur numérique, comme indiqué par le format "expression\_chaîne2".

### **Syntaxe**

```
to_number ( string_expression1 , string_expression2 )
```

### **to\_timestamp**

Convertit "expression\_chaîne1" en une valeur d'horodatage, comme indiqué par le format "expression\_chaîne2".

### **Syntaxe**

```
to_timestamp ( string_expression1 , string_expression2 )
```

## Fonctions mathématiques Greenplum

### **log**

Renvoie le logarithme en base 10 de "expression\_numérique1" ou le logarithme en base "expression\_numérique2".

### **Syntaxe**

```
log ( numeric_expression1 [ , numeric_expression2 ] )
```

### **ln**

Renvoie le logarithme naturel de "expression\_numérique1".

### **Syntaxe**

```
ln ( numeric_expression )
```

### **cbt**

Renvoie la racine cubique de "expression\_numérique1".

### **Syntaxe**

```
cbt ( numeric_expression )
```

### **pi**

Renvoie la constante du nombre Pi.

## Syntaxe

pi ( )

# Fonctions trigonométriques Greenplum

## acos

Renvoie l'arc cosinus de "expression\_numérique" en radians. L'arc cosinus est l'angle dont le cosinus est "expression\_numérique".

## Syntaxe

acos ( numeric\_expression )

## asin

Renvoie l'arc sinus de "expression\_numérique" en radians. L'arc sinus est l'angle dont le sinus est "expression\_numérique".

## Syntaxe

asin ( numeric\_expression )

## atan

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

## Syntaxe

atan ( numeric\_expression )

## atan2

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "expression\_numérique1" et "expression\_numérique2", en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique2"/"expression\_numérique1".

## Syntaxe

atan2 ( numeric\_expression1 ,numeric\_expression2 )

## cos

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

## Syntaxe

cos ( numeric\_expression )

## cot

Renvoie la cotangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

## Syntaxe

cot ( numeric\_expression )

## degrees

Renvoie une valeur en degrés où "expression\_numérique" représente un angle exprimé en radians.

## Syntaxe

degrees ( numeric\_expression )

## **radians**

Renvoie les radians où "expression\_numérique" correspond à un angle exprimé en degrés.

### **Syntaxe**

```
radians ( numeric_expression )
```

## **sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
sin ( numeric_expression )
```

## **tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
tan ( numeric_expression )
```

## **ascii**

Renvoie un nombre représentant la valeur de code ascii du caractère le plus à gauche dans "expression\_chaine" ; par exemple, ascii ( 'A' ) est 65.

### **Syntaxe**

```
ascii ( string_expression )
```

## **chr**

Renvoie le caractère dont la valeur de code ASCII est spécifiée par "expression\_entier". La valeur de "expression\_entier" doit être comprise entre 0 et 255.

### **Syntaxe**

```
chr ( integer_expression )
```

## **current\_database**

Renvoie le nom de la base de données en cours.

### **Syntaxe**

```
current_database ()
```

## **current\_schema**

Renvoie le nom du schéma en cours.

### **Syntaxe**

```
current_schema ()
```

## **{current\_user}**

### **Syntaxe**

```
{current_user}
```

## **{session\_user}**

### **Syntaxe**

{session\_user}

## **translate**

Renvoie la chaîne "expression\_chaine1" dans laquelle chaque occurrence de chaque caractère dans "expression\_chaine2" est remplacée par le caractère correspondant dans "expression\_chaine3".

### **Syntaxe**

translate ( string\_expression1 , string\_expression2 ,  
string\_expression3 )

## **date\_trunc**

Renvoie l'horodatage en fonction de la précision spécifiée.

### **Syntaxe**

date\_trunc ( string\_expression , timestamp\_expression )

## **version**

Renvoie la valeur de chaîne de la version de base de données.

### **Syntaxe**

version ( )

---

## **Informix**

### **Fonctions mathématiques Informix**

#### **log10**

Renvoie le logarithme en base 10 de "expression\_numérique".

#### **Syntaxe**

log10 ( numeric\_expression )

#### **logn**

Renvoie le logarithme naturel de "expression\_numérique".

#### **Syntaxe**

logn ( numeric\_expression )

#### **root**

Renvoie la valeur racine de "expression\_numérique1". Nécessite au moins un argument numérique (l'argument en radians). Si "expression\_numérique1" est la seule expression fournie, "expression\_numérique2" prend la valeur 2 par défaut. La valeur de "expression\_numérique2" doit être différente de zéro.

#### **Syntaxe**

root ( numeric\_expression1 [ , numeric\_expression2 ] )

## Fonctions trigonométriques Informix

### **acos**

Renvoie l'arc cosinus de "expression\_numérique" en radians. L'arc cosinus est l'angle dont le cosinus est "expression\_numérique".

### **Syntaxe**

```
acos ( numeric_expression )
```

### **asin**

Renvoie l'arc sinus de "expression\_numérique" en radians. L'arc sinus est l'angle dont le sinus est "expression\_numérique".

### **Syntaxe**

```
asin ( numeric_expression )
```

### **atan**

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

### **Syntaxe**

```
atan ( numeric_expression )
```

### **atan2**

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "expression\_numérique1" et "expression\_numérique2", en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique1".

### **Syntaxe**

```
atan2 ( numeric_expression1 , numeric_expression2 )
```

### **cos**

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

### **sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
sin ( numeric_expression )
```

### **tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
tan ( numeric_expression )
```

## **cardinality**

Renvoie le nombre d'éléments d'une colonne de collection (SET, MULTISSET, LIST).

## Syntaxe

cardinality ( string\_expression )

## char\_length

Renvoie le nombre de caractères logiques dans "expression\_chaine". Le nombre de caractères logiques peut être différent du nombre d'octets dans certains paramètres régionaux d'Asie orientale.

## Syntaxe

char\_length ( string\_expression )

## concat

Renvoie une chaîne qui correspond à la concaténation ou à la jointure des chaînes "expression\_chaine1" et "expression\_chaine2".

## Syntaxe

concat ( string\_expression1 , string\_expression2 )

## Exemple

```
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].  
[Sales staff].[Last name] )
```

Résultat

Renvoie le prénom et le nom ; par exemple, Bob Smith.

## date

Renvoie la valeur de date contenue dans "expression\_chaine", "expression\_date" ou "expression\_entier".

## Syntaxe

date ( string\_expression|date\_expression|integer\_expression )

## day

Renvoie un nombre entier qui représente le jour du mois (1 à 31).

## Syntaxe

day ( date\_expression )

## extend

Ajuste le degré de précision d'une expression de date-heure ou de date.

L'expression ne peut pas être une représentation entre guillemets de la valeur de date. Si le premier et le dernier qualificatifs ne sont pas spécifiés, les qualificatifs par défaut sont ceux de l'expression "year to fraction" (3). Si l'expression contient des zones qui ne sont pas définies par les qualificatifs, celles-ci sont supprimées. Si le premier qualificatif définit une zone plus longue (ou plus significative) que celle de l'expression, les valeurs renvoyées par la fonction en cours sont insérées dans les nouvelles zones. Si le dernier qualificatif définit une zone plus petite (ou moins significative) que celle de l'expression, des valeurs constantes sont insérées dans les nouvelles zones. La valeur 1 est insérée dans les zones MONTH ou DAY manquantes, alors que la valeur 0 est insérée dans les zones HOUR à FRACTION manquantes.

## Syntaxe

```
extend ( date_expression , ' { ' year to second ' } ' )
```

## Exemple

```
extend ( some_date_column , { year to second } )
```

## hex

Renvoie le codage hexadécimal de "expression\_entier".

## Syntaxe

```
hex ( integer_expression )
```

## initcap

Renvoie "expression\_chaine" avec la mise en forme suivante : la première lettre de chaque mot est en majuscule et les autres lettres sont en minuscules. Un mot commence après un caractère autre qu'une lettre. Ainsi, outre un espace vide, des symboles tels que des virgules, des points et des deux-points peuvent précéder un mot.

## Syntaxe

```
initcap ( string_expression )
```

## length

Renvoie le nombre d'octets de "expression\_chaine", en excluant les espaces blancs de fin de chaîne. Pour une chaîne "expression\_chaine" d'octet ou de texte, length renvoie le nombre total d'octets, y compris les espaces blancs de fin de chaîne.

## Syntaxe

```
length ( string_expression )
```

## lpad

Renvoie la chaîne "expression\_chaine1" précédée d'une chaîne "expression\_chaine2" d'une longueur totale de "expression\_entier" caractères. La séquence de "expression\_chaine2" est répétée autant de fois que nécessaire pour que la chaîne renvoyée compte "expression\_entier" caractères.

## Syntaxe

```
lpad ( string_expression1 , integer_expression , string_expression2 )
```

## mdy

Renvoie une valeur de type date avec trois expressions ayant pour résultat des nombres entiers qui correspondent au mois (expression\_entier1), au jour (expression\_entier2) et à l'année (expression\_entier).

## Syntaxe

```
mdy ( integer_expression1 , integer_expression2 ,  
integer_expression3 )
```

## month

Renvoie un nombre entier correspondant à la partie mois de "expression\_date".



## Syntaxe

month ( date\_expression )

## nvl

Renvoie la valeur de "expression1" si "expression1" n'est pas NULL. Si "expression1" est NULL, la valeur de "expression2" est renvoyée.

## Syntaxe

nvl ( expression1 , expression2 )

## Exemple

nvl ( [Unit sale price] , [Unit price] )

Résultat

Renvoie le prix unitaire de vente ou le prix unitaire si le premier est nul.

## octet\_length

Renvoie le nombre d'octets de "expression\_chaine", en incluant les espaces blancs de fin de chaîne.

## Syntaxe

octet\_length ( string\_expression )

## replace

Renvoie une instance de "expression\_chaine1" dans laquelle chaque occurrence de "expression\_chaine2" est remplacée par "expression\_chaine3". Si l'option "expression\_chaine3" est omise, chaque occurrence de "expression\_chaine2" est omise dans la chaîne renvoyée.

## Syntaxe

replace ( string\_expression1 , string\_expression2 [ ,  
string\_expression3 ] )

## Exemple

replace ( [Sales (query)].[Products].[Product line code] , - )

Résultat

Renvoie tous les codes des lignes de produits sans le caractère "-".

## round

Renvoie la valeur arrondie de "expression\_numérique". Si vous omettez "expression\_entier", la valeur est arrondie à zéro décimale ou à l'unité près. L'intervalle de 32 caractères numériques (+ et -) fait référence à la valeur décimale entière. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

## Syntaxe

round ( numeric\_expression [ , integer\_expression ] )

## Exemple

```
round (125, -1)
```

Résultat

130

## rpad

Renvoie la chaîne "expression\_chaine1" suivie d'une chaîne "expression\_chaine2" d'une longueur totale de "expression\_entier" caractères. La séquence de "expression\_chaine2" est répétée autant de fois que nécessaire pour que la chaîne renvoyée compte "expression\_entier" caractères.

## Syntaxe

```
rpad ( string_expression1 , integer_expression , string_expression2 )
```

## substr

Renvoie la sous-chaîne de "expression\_chaine" qui débute à la position "expression1\_entier" et dont la longueur est "expression2\_entier". Le premier caractère de "expression\_chaine" occupe la position 1. Si vous omettez "expression2\_entier", renvoie la sous-chaîne de "expression\_chaine" qui débute à la position "expression1\_entier" et se termine à la fin de "expression\_chaine".

## Syntaxe

```
substr ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

## Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## to\_char

Renvoie la chaîne de caractères "expression\_date" avec le formatage défini par "expression\_chaine". Vous ne pouvez utiliser cette fonction qu'avec des types de données prédéfinis.

## Syntaxe

```
to_char ( date_expression [ , string_expression ] )
```

## to\_date

Renvoie "expression\_chaine1" sous forme de date selon le format de date que vous spécifié dans "expression\_chaine2". Lorsque la valeur de "expression\_chaine1" est nulle, une valeur nulle est renvoyée.

## Syntaxe

```
to_date ( string_expression1 , string_expression2 )
```

## **trunc**

Renvoie la valeur tronquée de "expression\_numérique". Si vous omettez "expression\_entier", "expression\_numérique" est tronquée à zéro décimale ou à l'unité près. L'intervalle de 32 caractères numériques (+ et -) fait référence à la valeur décimale entière.

### **Syntaxe**

```
trunc ( numeric_expression [ , integer_expression ] )
```

## **weekday**

Renvoie un nombre entier qui représente le jour de la semaine de "expression\_date". Zéro (0) représente dimanche, un (1) représente lundi, etc.

### **Syntaxe**

```
weekday ( date_expression )
```

## **year**

Renvoie un nombre entier à quatre chiffres qui représente la partie année de "expression\_date".

### **Syntaxe**

```
year ( date_expression )
```

---

## **MS Access**

### **Fonctions de conversion MS Access**

#### **cast\_decimal**

Renvoie la valeur de "expression" convertie en une valeur décimale.

#### **Syntaxe**

```
cast_decimal ( expression )
```

#### **cast\_float**

Renvoie la valeur de "expression" convertie en une variable flottante.

#### **Syntaxe**

```
cast_float ( expression )
```

#### **cast\_integer**

Renvoie la valeur de "expression" convertie en un entier.

#### **Syntaxe**

```
cast_integer ( expression )
```

#### **Exemple**

```
cast_integer ( 84.95 )
```

Résultat

84

**cast\_numeric**

Renvoie la chaîne "expression\_chaîne" convertie en une valeur numérique.

**Syntaxe**

```
cast_numeric ( string_expression )
```

**cast\_real**

Renvoie la valeur de "expression" convertie en une valeur réelle.

**Syntaxe**

```
cast_real ( expression )
```

**cast\_smallint**

Renvoie la chaîne "expression" convertie en un chiffre entier.

**Syntaxe**

```
cast_smallint ( expression )
```

**cast\_varchar**

Renvoie la valeur de "expression" convertie en une zone de caractères variables.

**Syntaxe**

```
cast_varchar ( expression )
```

## Fonctions mathématiques MS Access

**log**

Renvoie le logarithme naturel de "expression\_numérique".

**Syntaxe**

```
log ( numeric_expression )
```

**rand**

Renvoie un nombre aléatoire en utilisant "expression\_entier" comme valeur de départ.

**Syntaxe**

```
rand ( integer_expression )
```

## Fonctions trigonométriques MS Access

**atan**

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

**Syntaxe**

```
atan ( numeric_expression )
```

**cos**

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

```
cos ( numeric_expression )
```

**sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

```
sin ( numeric_expression )
```

**tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

```
tan ( numeric_expression )
```

**ascii**

Renvoie la valeur de code ascii du caractère le plus à gauche dans "expression\_chaine".

**Syntaxe**

```
ascii ( string_expression )
```

**ceiling**

Renvoie le plus petit entier supérieur ou égal à "expression\_numérique".

**Syntaxe**

```
ceiling ( numeric_expression )
```

**chr**

Renvoie le caractère dont la valeur de code ASCII est spécifiée par "expression\_entier". La valeur de "expression\_entier" doit être comprise entre 0 et 255.

**Syntaxe**

```
chr ( integer_expression )
```

**concat**

Renvoie une chaîne qui correspond à la concaténation ou à la jointure des chaînes "expression\_chaine1" et "expression\_chaine2".

**Syntaxe**

```
concat ( string_expression1 , string_expression2 )
```

**Exemple**

```
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].  
[Sales staff].[Last name] )
```

Résultat

Renvoie le prénom et le nom ; par exemple, Bob Smith.

## curdate

Renvoie une valeur de date représentant la date en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

### Syntaxe

```
curdate ( )
```

## curtime

Renvoie une valeur d'heure représentant l'heure en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

### Syntaxe

```
curtime ( )
```

## dayname

Renvoie une chaîne de caractères contenant le nom du jour propre à la source de données (par exemple, Sunday à Saturday ou Sun. à Sat. pour une source de données en anglais, ou Sonntag à Samstag pour une source de données en allemand) pour la partie jour de "expression\_date".

### Syntaxe

```
dayname ( date_expression )
```

## dayofmonth

Renvoie le jour du mois (1-31) à partir de "expression\_date". Renvoie la zone des jours (entier signé) à partir de "expression\_intervalle".

### Syntaxe

```
dayofmonth ( date_expression | interval_expression )
```

## dayofweek

Renvoie le jour de la semaine dans "expression\_date" sous forme d'un entier (1-7), où 1 représente le lundi.

### Syntaxe

```
dayofweek ( date_expression )
```

## dayofyear

Renvoie le jour de l'année dans "expression\_date" sous forme d'un entier (1-366).

### Syntaxe

```
dayofyear ( date_expression )
```

## hour

Renvoie l'heure à partir de "time\_expression" sous la forme d'un entier compris entre 0 (minuit) et 23 (23:00).

### Syntaxe

```
hour ( time_expression )
```

## **instr**

Recherche la première occurrence de "expression\_chaine2" dans "expression\_chaine1" et renvoie un entier spécifiant la position de "expression\_chaine2". "expression\_entier1" définit la position de début pour la recherche. Si "expression\_entier1" est omis, la recherche débute à la position du premier caractère de "expression\_chaine1". "expression\_entier2" spécifie le type de comparaison de chaîne. "expression\_entier1" est requis si "expression\_entier2" est spécifié.

### **Syntaxe**

```
instr ( [ integer_expression1 , ] string_expression1 ,  
       string_expression2 [ , integer_expression2 ] )
```

## **lcase**

Convertit en minuscules tous les caractères en majuscules de "expression\_chaine".

### **Syntaxe**

```
lcase ( string_expression )
```

## **left**

Renvoie les "expression\_entier" caractères situés le plus à gauche dans "expression\_chaine".

### **Syntaxe**

```
left ( string_expression , integer_expression )
```

### **Exemple**

```
left ( [Sales (query)].[Sales staff].[Last name] , 3 )
```

Résultat

Renvoie les trois premiers caractères de chaque nom.

## **length**

Renvoie le nombre de caractères contenus dans "expression\_chaine", en excluant les espaces blancs à droite et le caractère de fin de chaîne.

### **Syntaxe**

```
length ( string_expression )
```

## **locate**

Renvoie la position de début de la première occurrence de "expression\_chaine1" dans "expression\_chaine2". La recherche débute à la position "expression\_entier" de "expression\_chaine2". Le premier caractère dans une chaîne figure en position 1. Si "expression\_chaine1" est introuvable, la valeur zéro est renvoyée.

### **Syntaxe**

```
locate ( string_expression1 , string_expression2 [ ,  
         integer_expression ] )
```

## **ltrim**

Renvoie "expression\_chaine" après avoir supprimé les espaces à gauche.

## Syntaxe

`ltrim ( string_expression )`

## minute

Renvoie la minute (nombre entier compris entre 0 et 59) contenue dans "time\_expression".

## Syntaxe

`minute ( time_expression )`

## month

Renvoie le mois (entier compris entre 1 et 12) contenu dans "expression\_date".

## Syntaxe

`month ( date_expression )`

## monthname

Renvoie une chaîne de caractères contenant le nom du mois propre à la source de données pour la partie mois de "expression\_date" (par exemple, January à December ou Jan. à Dec. pour une source de données en anglais ou Janvier à Décembre pour une source de données en français).

## Syntaxe

`monthname ( date_expression )`

## Exemple

`monthname ( 2005-11-01 )`

Résultat

Novembre

## now

Renvoie une valeur valeur date-heure représentant la date et l'heure en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

## Syntaxe

`now ( )`

## position

Renvoie la position de début de "expression1\_chaîne" dans "expression2\_chaîne". Le premier caractère d'une chaîne se trouve en position 1.

## Syntaxe

`position ( string_expression1 , string_expression2 )`

## quarter

Renvoie le trimestre contenu dans "expression\_date" sous forme d'un chiffre (1 à 4), où 1 représente les dates comprises entre le 1er janvier et le 31 mars.

## Syntaxe

`quarter ( date_expression )`



## right

Renvoie les "expression\_entier" caractères situés le plus à droite dans "expression\_chaine".

### Syntaxe

```
right ( string_expression , integer_expression )
```

## round

Renvoie l'expression "expression\_numérique" arrondie à la valeur la plus proche avec "expression\_entier" positions à droite du séparateur décimal. Si "expression\_entier" est une valeur négative, "expression\_numérique" est arrondi à la valeur absolue la plus proche avec "expression\_entier" positions à gauche du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression , integer_expression )
```

### Exemple

```
round (125, -1)
```

Résultat

130

## rtrim

Renvoie "expression\_chaine" après avoir supprimé les espaces de fin de chaîne.

### Syntaxe

```
rtrim ( string_expression )
```

### Exemple

```
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à la fin de ces derniers.

## sign

Renvoie un indicateur du signe de la chaîne "expression\_numérique" : +1 si elle est positive, 0 si elle est nulle et -1 si elle est négative.

### Syntaxe

```
sign ( numeric_expression )
```

## space

Renvoie une chaîne composée de "expression\_entier" espaces.

### Syntaxe

```
space ( integer_expression )
```

## substr

Renvoie la sous-chaîne de "expression\_chaîne" qui débute à la position "expression1\_entier" et dont la longueur est "expression2\_entier". Le premier caractère de "expression\_chaîne" se trouve en position 1.

### Syntaxe

```
substr ( string_expression , integer_expression1 ,  
        integer_expression2 )
```

### Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## sous-chaîne

Renvoie la sous-chaîne de "expression\_chaîne" qui débute à la position "expression1\_entier" et dont la longueur est "expression2\_entier". Le premier caractère de "expression\_chaîne" se trouve en position 1.

### Syntaxe

```
substring ( string_expression , integer_expression1 ,  
           integer_expression2 )
```

### Exemple

```
substring ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## truncate

Renvoie "expression\_chaîne" après avoir supprimé les espaces de fin de chaîne.

### Syntaxe

```
truncate ( string_expression )
```

## ucase

Convertit tous les caractères minuscules de "expression\_chaîne" en majuscules.

### Syntaxe

```
ucase ( string_expression )
```

## week

Renvoie la semaine de l'année contenue dans "expression\_date" sous forme d'un entier (1-53), où 1 représente la première semaine de l'année.

### Syntaxe

```
week ( date_expression )
```

## year

Renvoie l'année contenue dans "expression\_date".

### Syntaxe

```
year ( date_expression )
```

---

## MySQL

### Chaîne MySQL

#### lpad

Renvoie "expression\_chaine1" complétée à droite par "expression\_entier" occurrences de "expression\_chaine2". Si la longueur de "expression\_chaine1" dépasse "expression\_entier", la partie appropriée de "expression\_chaine1" est renvoyée.

#### Syntaxe

```
lpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

#### ltrim

Renvoie la chaîne "expression\_chaine1" dans laquelle les premiers caractères sont supprimés jusqu'au premier caractère qui n'appartient pas à "expression\_chaine2"; par exemple, ltrim ( 'xyxXxyAB' , 'xy' ) renvoie XxyAB.

#### Syntaxe

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

#### hex

Renvoie "expression\_numérique1" sous forme de chaîne hexadécimale.

#### Syntaxe

```
hex ( numeric_expression1 )
```

#### repeat

Renvoie la chaîne "expression\_chaine" répétée le nombre de fois déterminé par "expression\_numérique1".

#### Syntaxe

```
repeat ( string_expression , numeric_expression1 )
```

#### replace

Renvoie "expression\_chaine" après avoir remplacé "expression\_chaine2" par "expression\_chaine3".

#### Syntaxe

```
replace ( string_expression , string_expression2 , string_expression3 )
```

#### reverse

Renvoie l'expression "expression\_chaine" inversée.

#### Syntaxe

```
reverse ( string_expression )
```

**right**

Renvoie les "expression\_numérique" caractères situés le plus à droite dans "expression1\_chaîne".

**Syntaxe**

```
right ( string_expression1 , numeric_expression )
```

**rpad**

Renvoie "expression\_chaîne1" complétée à droite par "expression\_entier" occurrences de "expression\_chaîne2". Si la longueur de "expression\_chaîne1" dépasse "expression\_entier", la partie appropriée de "expression\_chaîne1" est renvoyée. Si "expression\_chaîne2" n'est pas spécifiée, des espaces sont utilisés.

**Syntaxe**

```
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

**rtrim**

Renvoie la chaîne "expression\_chaîne1" dans laquelle les caractères de fin ont été supprimés après le dernier caractère qui ne figure pas dans "expression\_chaîne2" ; par exemple, rtrim ( 'ABxXxyx' , 'xy' ) renvoie ABxX. Si "expression\_chaîne2" n'est pas spécifiée, les caractères espace de fin sont supprimés.

**Syntaxe**

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

**soundex**

Renvoie une chaîne soundex de "expression\_chaîne1".

**Syntaxe**

```
soundex ( string_expression1 )
```

## Fonctions mathématiques MySQL

**log**

Renvoie le logarithme en base 10 de "expression\_numérique1" ou le logarithme en base "expression\_numérique2".

**Syntaxe**

```
log ( numeric_expression )
```

**ln**

Renvoie le logarithme naturel de "expression\_numérique1".

**Syntaxe**

```
ln ( numeric_expression )
```

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

```
pi ( )
```

## Fonctions trigonométriques MySQL

### **acos**

Renvoie l'arc cosinus de "expression\_numérique" en radians. L'arc cosinus est l'angle dont le cosinus est "expression\_numérique".

### **Syntaxe**

```
acos ( numeric_expression )
```

### **asin**

Renvoie l'arc sinus de "expression\_numérique" en radians. L'arc sinus est l'angle dont le sinus est "expression\_numérique".

### **Syntaxe**

```
asin ( numeric_expression )
```

### **atan**

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

### **Syntaxe**

```
atan ( numeric_expression )
```

### **atan2**

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "expression\_numérique1" et "expression\_numérique2", en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique2"/"expression\_numérique1".

### **Syntaxe**

```
atan2 ( numeric_expression1 ,numeric_expression2 )
```

### **cos**

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

### **cot**

Renvoie la cotangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

### **Syntaxe**

```
cot ( numeric_expression )
```

### **degrees**

Renvoie une valeur en degrés où "expression\_numérique" représente un angle exprimé en radians.

### **Syntaxe**

```
degrees ( numeric_expression )
```

### **radians**

Renvoie les radians où "expression\_numérique" correspond à un angle exprimé en degrés.

**Syntaxe**

radians ( numeric\_expression )

**sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

sin ( numeric\_expression )

**tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

**Syntaxe**

tan ( numeric\_expression )

**ascii**

Renvoie un nombre représentant la valeur de code ASCII du caractère le plus à gauche dans "expression\_chaine" ; par exemple, ascii ( 'A' ) est 65.

**Syntaxe**

ascii ( string\_expression )

**database**

Renvoie le nom de la base de données en cours.

**Syntaxe**

database ()

**schema**

Renvoie le nom du schéma en cours.

**Syntaxe**

schema ()

**session\_user**

Renvoie le nom d'utilisateur renvoyé par le client.

**Syntaxe**

session\_user ()

**system\_user**

Renvoie le nom d'utilisateur renvoyé par le client.

**Syntaxe**

system\_user ()

**version**

Renvoie la valeur de chaîne de la version de base de données.

## Syntaxe

version ()

---

## Netezza

### Fonctions mathématiques Netezza

#### log

Renvoie le logarithme de "expression\_numérique2" en base "expression\_numérique1". Lorsque l'argument facultatif "expression\_numérique1" n'est pas défini, la base 10 est utilisée.

#### Syntaxe

```
log ( [numeric_expression1 ,] numeric_expression2 )
```

### Fonctions trigonométriques Netezza

#### acos

Renvoie l'arc cosinus de "expression\_numérique" en radians. L'arc cosinus est l'angle dont le cosinus est "expression\_numérique".

#### Syntaxe

```
acos ( numeric_expression )
```

#### asin

Renvoie l'arc sinus de "expression\_numérique" en radians. L'arc sinus est l'angle dont le sinus est "expression\_numérique".

#### Syntaxe

```
asin ( numeric_expression )
```

#### atan

Renvoie l'arc tangente de "expression\_numérique" en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique".

#### Syntaxe

```
atan ( numeric_expression )
```

#### atan2

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "expression\_numérique1" et "expression\_numérique2", en radians. L'arc tangente est l'angle dont la tangente est "expression\_numérique2"/"expression\_numérique1".

#### Syntaxe

```
atan2 ( numeric_expression1 , numeric_expression2 )
```

#### cos

Renvoie le cosinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

#### Syntaxe

```
cos ( numeric_expression )
```

### **degrees**

Renvoie une valeur en degrés où "expression\_numérique" représente un angle exprimé en radians.

#### **Syntaxe**

```
degrees ( numeric_expression )
```

### **radians**

Renvoie les radians où "expression\_numérique" correspond à un angle exprimé en degrés.

#### **Syntaxe**

```
radians ( numeric_expression )
```

### **sin**

Renvoie le sinus de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

#### **Syntaxe**

```
sin ( numeric_expression )
```

### **tan**

Renvoie la tangente de "expression\_numérique" où "expression\_numérique" est un angle exprimé en radians.

#### **Syntaxe**

```
tan ( numeric_expression )
```

## **Netezza Fuzzy**

### **le\_dst**

Renvoie une valeur indiquant le degré de différence entre deux chaînes d'entrée, calculé d'après l'algorithme de distance de Levenshtein.

#### **Syntaxe**

```
le_dst ( string_expression1 , string_expression2 )
```

### **dle\_dst**

Renvoie une valeur indiquant le degré de différence entre deux chaînes d'entrée, calculé d'après l'algorithme de distance de Damerau-Levenshtein.

#### **Syntaxe**

```
dle_dst ( string_expression1 , string_expression2 )
```

## **Fonctions phonétiques Netezza**

### **nysiis**

Renvoie une représentation Soundex de "expression\_chaine" en utilisant la variante NYSIIS (New York State Identification and Intelligence System) de Soundex.

#### **Syntaxe**

```
nysiis ( string_expression )
```

### **dbl\_mp**

Renvoie une valeur 32 bits composite de "expression\_chaine".



**Syntaxe**

`dbl_mp ( string_expression )`

**pri\_mp**

Renvoie la chaîne Metaphone primaire à 4 caractères à partir de la valeur de "expression\_numérique" renvoyée par `dbl_mp`.

**Syntaxe**

`pri_mp ( numeric_expression )`

**sec\_mp**

Renvoie la chaîne Metaphone secondaire à 4 caractères à partir de la valeur de "expression\_numérique" renvoyée par `dbl_mp`.

**Syntaxe**

`sec_mp ( numeric_expression )`

**score\_mp**

Renvoie un score de correspondance indiquant la proximité entre "expression\_numérique" et "expression\_numérique2".

**Syntaxe**

`score_mp ( numeric_expression , numeric_expression2 ,  
numeric_expression3 , numeric_expression4 , numeric_expression5 ,  
numeric_expression6 )`

**ascii**

Renvoie un nombre représentant la valeur de code ASCII du caractère le plus à gauche dans "expression\_chaine" ; par exemple, `ascii ( 'A' )` est 65.

**Syntaxe**

`ascii ( string_expression )`

**chr**

Renvoie le caractère dont la valeur de code ASCII est spécifiée par "expression\_entier". La valeur de "expression\_entier" doit être comprise entre 0 et 255.

**Syntaxe**

`chr ( integer_expression )`

**decode**

Compare "expr" à chaque valeur de recherche, une par une. Si "expr" correspond à une valeur de recherche, le résultat correspondant est renvoyé. Si aucune correspondance n'est trouvée, la valeur "default" est renvoyée. Si la valeur "default" est omise, une valeur nulle est renvoyée.

**Syntaxe**

`decode ( expr , search , result [ , search , result]... [ , default] )`

## initcap

Renvoie "expression\_chaine" avec la mise en forme suivante : la première lettre de chaque mot est en majuscule et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

### Syntaxe

```
initcap ( string_expression )
```

## instr

Recherche la "expression\_entier2" occurrence de "expression\_chaine2" dans "expression\_chaine1" à partir de la position "expression\_entier1". Si la valeur de "expression\_entier1" est négative, la recherche s'effectue vers l'arrière à partir de la fin de "expression\_chaine1". Renvoie un entier indiquant la position de "expression\_chaine2".

### Syntaxe

```
instr ( string_expression1 , string_expression2 [ ,  
integer_expression1 [ , integer_expression2 ] ] )
```

## lpad

Renvoie "expression1\_chaine" complétée à droite par "expression\_entier" occurrences de "expression\_chaine2". Si la longueur de "expression\_chaine1" dépasse "expression\_entier", la partie appropriée de "expression\_chaine1" est renvoyée.

### Syntaxe

```
lpad ( string_expression1 , integer_expression [ , string_expression2  
] )
```

## ltrim

Renvoie la chaîne "expression\_chaine1" dans laquelle les premiers caractères sont supprimés jusqu'au premier caractère qui n'appartient pas à "expression\_chaine2" ; par exemple, ltrim ( 'xyxXxyAB' , 'xy' ) renvoie XxyAB.

### Syntaxe

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

## months\_between

Renvoie le nombre de mois entre "expression\_date1" et "expression\_date2". Si "expression\_date1" est postérieure à "expression\_date2", le résultat est un nombre positif. La fonction ne tient pas compte des jours et des heures résultant de la différence, c'est-à-dire que les mois ne sont pas arrondis, sauf si "expression\_date1" et "expression\_date2" représentent les derniers jours d'un mois.

### Syntaxe

```
months_between ( date_expression1 , date_expression2 )
```

## next\_day

Renvoie la date et l'heure du premier jour postérieur à "expression\_date-heure" dont le nom de jour de la semaine est indiqué par "expression\_chaine". Les valeurs renvoyées pour les heures, les minutes et les secondes sont les mêmes que celles de "expression\_date-heure".

## Syntaxe

`next_day ( datetime_expression , string_expression )`

## nvl

Renvoie la valeur de "expression1" si "expression1" n'est pas NULL. Si "expression1" est NULL, la valeur de "expression2" est renvoyée.

## Syntaxe

`nvl ( expression1 , expression2 )`

## Exemple

`nvl ( [Unit sale price] , 0 )`

Résultat

Renvoie le prix unitaire de vente ou 0 si le prix unitaire de vente est nul.

## round

Renvoie l'expression "expression\_numérique" arrondie à la valeur la plus proche avec "expression\_entier" positions à droite du séparateur décimal. Si "expression\_entier" est une valeur négative, "expression\_numérique" est arrondie à la valeur absolue la plus proche avec "expression\_entier" positions à gauche du séparateur décimal ; par exemple, `round (125, -1)` permet d'arrondir à 130.

## Syntaxe

`round ( numeric_expression [ , integer_expression ] )`

## rpad

Renvoie "expression\_chaine1" complétée à droite par "expression\_entier" occurrences de "expression\_chaine2". Si la longueur de "expression\_chaine1" dépasse "expression\_entier", la partie appropriée de "expression\_chaine1" est renvoyée. Si "expression\_chaine2" n'est pas spécifiée, des espaces sont utilisés.

## Syntaxe

`rpad ( string_expression1 , integer_expression [ , string_expression2 ] )`

## rtrim

Renvoie la chaîne "expression\_chaine1" dans laquelle les caractères de fin ont été supprimés après le dernier caractère qui ne figure pas dans "expression\_chaine2" ; par exemple, `rtrim ( 'ABxXyx' , 'xy' )` renvoie ABxX. Si "expression\_chaine2" n'est pas spécifiée, les caractères espace de fin sont supprimés.

## Syntaxe

`rtrim ( string_expression1 [ , string_expression2 ] )`

## substr

Renvoie la sous-chaîne de "expression\_chaine" qui débute à la position "expression1\_entier". Le premier caractère de "expression\_chaine" se trouve en position 1. L'élément "expression2\_entier" peut être utilisé pour sélectionner moins de caractères. Par défaut, il sélectionne les caractères en fin de chaîne.

## Syntaxe

```
substr ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

## {current\_db}

### Syntaxe

```
{current_db}
```

## {current\_user}

### Syntaxe

```
{current_user}
```

## {session\_user}

### Syntaxe

```
{session_user}
```

## to\_char

Renvoie "expression" sous forme de chaîne de type "expression\_chaîne". "expression" peut être une valeur de date ou une valeur numérique.

### Syntaxe

```
to_char ( expression [ , string_expression ] )
```

## to\_date

Convertit "expression1" en une valeur de date de type "expression\_chaîne2". Le premier argument "expression1" peut être de type chaîne ou numérique.

### Syntaxe

```
to_date ( expression1 , string_expression2 )
```

### Exemple

```
to_date ( '2013-02-27' , 'YYYY-MM-DD' )
```

Résultat

Renvoie une date avec la valeur '2013-02-27'.

## to\_number

Convertit "expression\_chaîne1" en une valeur numérique, comme indiqué par le format "expression\_chaîne2".

### Syntaxe

```
to_number ( string_expression1 , string_expression2 )
```

## translate

Renvoie la chaîne "expression\_chaîne1", dans laquelle toutes les occurrences de chaque caractère de "expression\_chaîne2" sont remplacées par le caractère correspondant dans "expression\_chaîne3".

### Syntaxe

```
translate ( string_expression1 , string_expression2 ,  
string_expression3 )
```

### date\_trunc

Tronque "expression\_date1" en une valeur, comme indiqué par le format "expression\_chaine1".

### Syntaxe

```
date_trunc ( string_expression1 , date_expression1 )
```

### trunc

Tronque les chiffres de "expression\_numérique1" avec "expression\_numérique2" comme niveau de précision.

### Syntaxe

```
trunc ( numeric_expression1 [ , numeric_expression2 ] )
```

### version

Renvoie la valeur "expression\_chaine1" correspondant à la version de base de données.

### Syntaxe

```
version ()
```

---

## Oracle

### Fonctions mathématiques Oracle

#### log

Renvoie le logarithme de "numeric\_expression2" en base "numeric\_expression1". Lorsque l'argument facultatif "expression\_numérique1" n'est pas défini, la base utilisée est la constante e (qui est approximativement égale à 2.71282).

#### Syntaxe

```
log ([ numeric_expression1 , ] numeric_expression2 )
```

### Fonctions trigonométriques Oracle

#### acos

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

#### Syntaxe

```
acos ( numeric_expression )
```

#### asin

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

#### Syntaxe

```
asin ( numeric_expression )
```

## **atan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

### **Syntaxe**

```
atan ( numeric_expression )
```

## **atan2**

Renvoie l'arc tangente des coordonnées x et y définies respectivement par "numeric\_expression1" et "numeric\_expression2", en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression2"/"numeric\_expression1".

### **Syntaxe**

```
atan2 ( numeric_expression1 ,numeric_expression2 )
```

## **cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

## **cosh**

Renvoie le cosinus hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cosh ( numeric_expression )
```

## **sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
sin ( numeric_expression )
```

## **sinh**

Renvoie le sinus hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
sinh ( numeric_expression )
```

## **tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
tan ( numeric_expression )
```

## **tanh**

Renvoie la tangente hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
tanh ( numeric_expression )
```

**add\_months**

Renvoie l'élément date-heure résultant de l'ajout de "integer\_expression" mois à "date\_expression".

**Syntaxe**

```
add_months ( date_expression , integer_expression )
```

**ascii**

Renvoie un nombre correspondant au code ASCII du premier caractère de "string\_expression".

**Syntaxe**

```
ascii ( string_expression )
```

**Exemple**

```
ascii ( 'A' )
```

Résultat

Renvoie '65'

**ceil**

Renvoie le plus petit entier supérieur ou égal à "numeric\_expression".

**Syntaxe**

```
ceil ( numeric_expression )
```

**char\_length**

Renvoie le nombre de caractères logiques contenus dans "string\_expression". Le nombre de caractères logiques peut être différent du nombre d'octets dans certaines langues d'Asie orientale.

**Syntaxe**

```
char_length ( string_expression )
```

**chr**

Renvoie le caractère qui correspond au code ASCII indiqué par "integer\_expression". La valeur de "integer\_expression" doit être comprise entre 0 et 255.

**Syntaxe**

```
chr ( integer_expression )
```

**concat**

Renvoie la chaîne résultant de la concaténation ou de la jointure de "string\_expression1" et "string\_expression2".

**Syntaxe**

```
concat ( string_expression1 , string_expression2 )
```

## Exemple

```
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].  
[Sales staff].[Last name] )
```

Résultat

Renvoie le prénom et le nom ; par exemple, Bob Smith.

## decode

Compare "expression" à chaque valeur de recherche, une par une. Si "expression" correspond à une recherche, le résultat correspondant est renvoyé. Si aucune correspondance n'est trouvée, la fonction renvoie la valeur par défaut "default". Si "default" est omis, la valeur renvoyée est Null.

### Syntaxe

```
decode ( expression , search , result [ , search , result ]... [ ,  
default ] )
```

## dump

Renvoie la représentation interne de "expression" au format "numeric\_expression1" en commençant par la position "numeric\_expression2" pour "numeric\_expression3" caractères.

### Syntaxe

```
dump ( expression [ , numeric_expression1 [ , numeric_expression2 [ ,  
numeric_expression3 ] ] ] )
```

## greatest

Renvoie la valeur la plus élevée de "expression\_list".

### Syntaxe

```
greatest ( expression_list )
```

## initcap

Renvoie l'expression de chaîne "string\_expression" mise en forme de la façon suivante : la première lettre de chaque mot est en majuscules et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

### Syntaxe

```
initcap ( string_expression )
```

## instr

Recherche "string\_expression1" en commençant par la position "integer\_expression1" pour l'occurrence "integer\_expression2" de "string\_expression2". Si la valeur de "integer\_expression1" est négative, la recherche se fait en amont à partir de la fin de "string\_expression1". Renvoie un entier indiquant la position de "string\_expression2".

### Syntaxe

```
instr ( string_expression1, string_expression2 [ ,  
integer_expression1 [ , integer_expression2 ] ] )
```



## **instrb**

Recherche "string\_expression1" en commençant par la position "integer\_expression1" pour l'occurrence "integer\_expression2" de "string\_expression2". Si la valeur de "integer\_expression1" est négative, la recherche se fait en amont à partir de la fin de "string\_expression1". Renvoie la position (numéro d'octet) dans laquelle "string\_expression2" a été trouvée.

### **Syntaxe**

```
instrb ( string_expression1, string_expression2 [ ,  
integer_expression1 [ , integer_expression2 ] ] )
```

## **least**

Renvoie la valeur la moins élevée de "expression\_list".

### **Syntaxe**

```
least ( expression_list )
```

## **length**

Renvoie le nombre de caractères contenus dans "string\_expression".

### **Syntaxe**

```
length ( string_expression )
```

## **lengthb**

Renvoie le nombre d'octets contenus dans "string\_expression".

### **Syntaxe**

```
lengthb ( string_expression )
```

## **lpad**

Renvoie "string\_expression1" complétée à gauche jusqu'à la longueur définie par "integer\_expression" avec les occurrences de "string\_expression2". Si la longueur de "string\_expression1" dépasse "integer\_expression", la partie appropriée de "string\_expression1" est renvoyée.

### **Syntaxe**

```
lpad ( string_expression1, integer_expression [ ,  
string_expression2 ] )
```

## **ltrim**

Renvoie "string\_expression1" en supprimant les caractères initiaux jusqu'au premier caractère ne figurant pas dans "string\_expression2".

### **Syntaxe**

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

### **Exemple**

```
ltrim ( 'xyxXxyAB' , 'xy' )
```

Résultat

XxyAB

## months\_between

Renvoie le nombre de mois compris entre "date\_expression1" et "date\_expression2". Si l'expression "date\_expression1" est ultérieure à "date\_expression2", le résultat sera positif. La fonction ne tient pas compte des jours et des heures résultant de la différence, c'est-à-dire que les mois ne sont pas arrondis, sauf si "date\_expression1" et "date\_expression2" sont les derniers jours du mois.

### Syntaxe

```
months_between ( date_expression1 , date_expression2 )
```

## new\_time

Renvoie la date-heure dans le nouveau fuseau horaire "new\_timezone" pour "datetime\_expression" dans l'ancien fuseau horaire "old\_timezone". Les valeurs de "old\_timezone" et "new\_timezone" peuvent être : 'AST', 'ADT', 'BST', 'BDT', 'CST', 'CDT', 'EST', 'EDT', 'HST', 'HDT', 'MST', 'MDT', 'NST', 'PST', 'PDT', 'YST' ou 'YDT'.

### Syntaxe

```
new_time ( datetime_expression , old_timezone , new_timezone )
```

## next\_day

Renvoie la date et l'heure du premier jour postérieur à "string\_expression" dont le nom de jour de la semaine est donné par "datetime\_expression". La valeur renvoyée a le même format que "datetime\_expression".

### Syntaxe

```
next_day ( datetime_expression , string_expression )
```

## nls\_initcap

Renvoie l'expression de chaîne "string\_expression1" mise en forme de la façon suivante : la première lettre de chaque mot est en majuscules et les autres lettres sont en minuscules. Cette fonction permet de faire précéder un mot d'un caractère autre qu'une lettre. Ainsi, en plus des espaces blancs, les symboles tels que les virgules, les points et les deux-points peuvent précéder un mot. La valeur de "string\_expression2" définit la séquence de tri.

### Syntaxe

```
nls_initcap ( string_expression1 [ , string_expression2 ] )
```

## nls\_lower

Renvoie "string\_expression1" avec toutes les lettres en minuscules. La valeur de "string\_expression2" définit la séquence de tri.

### Syntaxe

```
nls_lower ( string_expression1 [ , string_expression2 ] )
```

## nls\_upper

Renvoie "string\_expression1" avec toutes les lettres en majuscules. La valeur de "string\_expression2" définit la séquence de tri.

### Syntaxe

```
nls_upper ( string_expression1 [ , string_expression2 ] )
```

## **nvl**

Renvoie la valeur de "expression1" si "expression1" n'est pas NULL. Si "expression1" est NULL, la valeur de "expression2" est renvoyée.

### **Syntaxe**

```
nvl ( expression1 , expression2 )
```

### **Exemple**

```
nvl ( [Unit sale price] , 0 )
```

Résultat

Renvoie le prix unitaire de vente ou 0 si le prix unitaire de vente est nul.

## **replace**

Remplace toutes les occurrences de "string\_expression2" par "string\_expression3" dans "string\_expression1". Si l'expression de chaîne "string\_expression3" n'est pas définie, toutes les occurrences de "string\_expression2" sont supprimées.

### **Syntaxe**

```
replace ( string_expression1 , string_expression2 [ ,  
string_expression3 ] )
```

## **round**

Renvoie l'expression "numeric\_expression" arrondie à la valeur la plus proche avec "integer\_expression" positions à droite du séparateur décimal. Si "integer\_expression" est une valeur négative, "numeric\_expression" est arrondi à la valeur absolue la plus proche avec "integer\_expression" positions à gauche du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### **Syntaxe**

```
round ( numeric_expression [ , integer_expression ] )
```

### **Exemple**

```
round ( 125 , -1 )
```

Résultat

Renvoie 130

## **rpad**

Renvoie l'expression de chaîne "string\_expression1" complétée à droite par "integer\_expression" occurrences de "string\_expression2". Si la longueur de "string\_expression1" dépasse "integer\_expression", la partie appropriée de "string\_expression1" est renvoyée. Si l'expression de chaîne "string\_expression2" n'est pas indiquée, les occurrences de "string\_expression2" sont remplacées par des espaces.

### **Syntaxe**

```
rpad ( string_expression1 , integer_expression [ ,  
string_expression2 ] )
```

## rtrim

Renvoie "string\_expression1" en supprimant les caractères de fin jusqu'au dernier caractère ne figurant pas dans "string\_expression2". Si l'expression de chaîne "string\_expression2" n'est pas spécifiée, les caractères espace finaux sont supprimés.

### Syntaxe

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

### Exemple

```
rtrim ( 'ABxXxyx' , 'xy' )
```

Résultat

Renvoie 'ABxX'

## sign

Renvoie +1 si l'expression "numeric\_expression" est positive, 0 si elle est nulle, et -1 si elle est négative.

### Syntaxe

```
sign ( numeric_expression )
```

## soundex

Renvoie une chaîne de caractères constituant une représentation phonétique de "string\_expression".

### Syntaxe

```
soundex ( string_expression )
```

## substr

Renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1" et dont la longueur est de "integer\_expression2" caractères, ou qui se termine par "string\_expression" si l'expression "integer\_expression2" est omise. Le premier caractère de "string\_expression" occupe la position 1.

### Syntaxe

```
substr ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

### Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## substrb

Renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1" et se termine après "numeric\_expression2" octets. Le premier octet de l'expression de chaîne "string\_expression" se trouve en position 1. Si l'expression "numeric\_expression2" est omise, la fonction renvoie la sous-chaîne de "string\_expression" qui commence à la position "numeric\_expression1" et se termine par la fin de "string\_expression".

## Syntaxe

```
substrb ( string_expression , numeric_expression1 [ ,  
         numeric_expression2 ] )
```

## Exemple

```
substrb ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## {sysdate}

Renvoie une valeur de date-heure représentant la date et l'heure en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

## Syntaxe

```
{ sysdate }
```

## to\_char

Renvoie "expression" sous forme de chaîne de type "string\_expression". "expression" peut être une valeur de date ou une valeur numérique.

## Syntaxe

```
to_char ( expression [ , string_expression ] )
```

## to\_date

Convertit "expression1" en une valeur date-heure de type "expression\_chaine2". Le premier argument "expression1" peut être de type chaîne ou numérique. La valeur de "string\_expression3" définit les éléments du format, tels que la langue.

## Syntaxe

```
to_date ( expression1 [ , string_expression2 [ ,  
         string_expression3 ] ] )
```

## Exemple 1

```
to_date ( '2013-02-27 15:13:14' , 'YYYY-MM-DD HH24:MI:SS' )
```

Résultat

Renvoie une date et une heure avec la valeur '2013-02-27 15:13:14'.

## Exemple 2

```
to_date ( '2013-02-27 15:13:14' , 'YYYY-MM-DD HH24:MI:SS' ,  
         'NLS_DATE_LANGUAGE = American' )
```

Résultat

Renvoie une date et une heure avec la valeur '2013-02-27 15:13:14'.

## to\_number

Convertit "string\_expression1" en une valeur numérique de type "string\_expression2". La valeur de "string\_expression3" définit les éléments du format, tels que les informations relatives à la devise.

## Syntaxe

```
to_number ( string_expression1 , string_expression2 ,  
            string_expression3 )
```

## translate

Renvoie l'expression de chaîne "string\_expression1", dans laquelle chacune des occurrences d'un caractère de "string\_expression2" est remplacée par le caractère correspondant de "string\_expression3".

## Syntaxe

```
translate ( string_expression1 , string_expression2 ,  
            string_expression3 )
```

## trunc

Tronque "date\_expression" en fonction du format indiqué par "string\_expression". Par exemple, si "string\_expression" est "year", alors l'expression de date "date\_expression" est tronquée au premier jour de l'année.

## Syntaxe

```
trunc ( date_expression , string_expression )
```

## Exemple

```
trunc ( 2003-08-22 , 'year' )
```

Résultat

Renvoie 2003-01-01.

## trunc

Tronque les chiffres de "numeric\_expression1" avec "numeric\_expression2" comme niveau de précision.

## Syntaxe

```
trunc ( numeric_expression1 , numeric_expression2 )
```

## {user}

Renvoie le nom de l'utilisateur Oracle en cours.

## Syntaxe

```
{ user }
```

## vsize

Renvoie le nombre d'octets de la représentation interne de "expression". "expression" doit être une expression de chaîne.

## Syntaxe

```
vsize ( expression )
```

### Chaîne Paracel

#### **overlay**

Renvoie "string\_expression1" en remplaçant "string\_expression2" avec la position de caractère "numeric\_expression".

#### **Syntaxe**

```
overlay ( string_expression1 , string_expression2 ,  
numeric_expression1 [ , numeric_expression2 ] )
```

#### **ltrim**

Renvoie l'expression de chaîne "string\_expression1" dans laquelle les caractères de début sont supprimés jusqu'au premier caractère n'appartenant pas à "string\_expression2". Exemple : ltrim('xyxXxyAB', 'xy') donne 'XxyAB'.

#### **Syntaxe**

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

#### **replace**

Renvoie "string\_expression" en remplaçant "string\_expression2" par "string\_expression3".

#### **Syntaxe**

```
replace ( string_expression , string_expression2 , string_expression3  
)
```

#### **rtrim**

Renvoie "string\_expression1" dans laquelle les derniers caractères sont supprimés après le dernier caractère qui n'appartient pas à "string\_expression2". Exemple : rtrim ('ABxXxyx', 'xy') donne ABxX. Si "string\_expression2" n'est pas spécifiée, les caractères espace finaux sont supprimés.

#### **Syntaxe**

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

### Formatage des types de données Paracel

#### **to\_char**

Renvoie "expression" sous forme de chaîne de type "string\_expression". "Expression" peut être une valeur de date ou une valeur numérique.

#### **Syntaxe**

```
to_char ( expression , string_expression )
```

#### **to\_date**

Convertit "string\_expression1" en une valeur de date de type "string\_expression2".

#### **Syntaxe**

```
to_date ( string_expression1 , string_expression2 )
```

#### **to\_number**

Convertit "string\_expression1" en une valeur numérique de type "string\_expression2".

**Syntaxe**

`to_number ( string_expression1 , string_expression2 )`

## Fonctions mathématiques Paracel

**cbrt**

Renvoie la racine cubique de "numeric\_expression1".

**Syntaxe**

`cbrt ( numeric_expression )`

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

`pi ( )`

### current\_database

Renvoie le nom de la base de données en cours.

**Syntaxe**

`current_database ( )`

### current\_schema

Renvoie le nom du schéma en cours.

**Syntaxe**

`current_schema ( )`

### {current\_user}

**Syntaxe**

`{current_user}`

### {session\_user}

**Syntaxe**

`{session_user}`

### translate

Renvoie "expression1\_chaîne", dans laquelle chaque occurrence de caractère contenue dans "expression2\_chaîne" est remplacée par le caractère correspondant de "expression3\_chaîne".

**Syntaxe**

`translate ( string_expression1 , string_expression2 , string_expression3 )`

### version

Renvoie la valeur de chaîne de la version de base de données.

**Syntaxe**

`version ( )`



### Chaîne Postgres

#### **overlay**

Renvoie "string\_expression1" en remplaçant "string\_expression2" à partir de la position du caractère "numeric\_expression".

#### **Syntaxe**

```
overlay ( string_expression1 , string_expression2 ,  
numeric_expression1 [ , numeric_expression2 ] )
```

#### **btrim**

Renvoie expression\_chaîne1 après avoir supprimé la chaîne de caractères la plus longue dans "expression\_chaîne2".

#### **Syntaxe**

```
btrim ( string_expression1 [ , string_expression2 ] )
```

#### **initcap**

Renvoie "string\_expression" mise en forme de la façon suivante : la première lettre de chaque mot est en majuscules et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

#### **Syntaxe**

```
initcap ( string_expression )
```

#### **lpad**

Renvoie l'expression de chaîne "string\_expression1" complétée par "integer\_expression" occurrences de "string\_expression2". Si la longueur de "string\_expression1" dépasse "integer\_expression", la partie appropriée de "string\_expression1" est renvoyée.

#### **Syntaxe**

```
lpad ( string_expression1 , integer_expression [ , string_expression2  
] )
```

#### **ltrim**

Renvoie l'expression de chaîne "string\_expression1" dans laquelle les caractères de début sont supprimés jusqu'au premier caractère n'appartenant pas à "string\_expression2". Exemple : ltrim('xyxXxyAB', 'xy') donne 'XxyAB'.

#### **Syntaxe**

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

#### **md5**

Renvoie le hachage MD5 de "string\_expression1".

#### **Syntaxe**

```
md5 ( string_expression1 )
```

#### **to\_hex**

Renvoie une représentation sous forme de chaîne hexadécimale de "numeric\_expression1".

**Syntaxe**

```
to_hex ( numeric_expression1 )
```

**repeat**

Renvoie la chaîne "string\_expression" répétée le nombre de fois déterminé par "numeric\_expression1".

**Syntaxe**

```
repeat ( string_expression , numeric_expression1 )
```

**replace**

Renvoie "string\_expression" avec "string\_expression2" remplacé par "string\_expression3".

**Syntaxe**

```
replace ( string_expression , string_expression2 , string_expression3)
```

**rpad**

Renvoie l'expression de chaîne "string\_expression1" complétée à droite par "integer\_expression" occurrences de "string\_expression2". Si la longueur de "string\_expression1" dépasse "integer\_expression", la partie appropriée de "string\_expression1" est renvoyée. Si "string\_expression2" n'est pas spécifié, des espaces sont utilisés.

**Syntaxe**

```
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

**rtrim**

Renvoie "string\_expression1" dans laquelle les derniers caractères sont supprimés après le dernier caractère qui n'appartient pas à "string\_expression2". Exemple : rtrim ('ABxXxyx', 'xy') donne ABxX. Si l'expression "string\_expression2" n'est pas spécifiée, les caractères espace finaux sont supprimés.

**Syntaxe**

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

**split\_part**

Renvoie "numeric\_expression" en séparant "string\_expression1" dans "string\_expression2".

**Syntaxe**

```
split_part ( string_expression1 , string_expression2 ,  
numeric_expression )
```

## Formatage des types de données Postgres

**to\_char**

Renvoie "expression" sous forme de chaîne de type "string\_expression". "Expression" peut être une valeur de date ou une valeur numérique.

**Syntaxe**

```
to_char ( expression , string_expression )
```

**to\_date**

Convertit "string\_expression1" en une valeur de date de type "string\_expression2".

**Syntaxe**

to\_date ( string\_expression1 , string\_expression2 )

**to\_number**

Convertit "string\_expression1" en une valeur numérique de type "string\_expression2".

**Syntaxe**

to\_number ( string\_expression1 , string\_expression2 )

**to\_timestamp**

Convertit "string\_expression1" en une valeur d'horodatage au format "string\_expression2". Autre syntaxe : to\_timestamp ( numeric-expression )  
Convertit le timestamp Unix en date.

**Syntaxe**

to\_timestamp ( string\_expression1 , string\_expression2 )

## Fonctions mathématiques Postgres

**log**

Renvoie le logarithme en base 10 de "numeric\_expression1" ou le logarithme en base "numeric\_expression2".

**Syntaxe**

log ( numeric\_expression1 [ , numeric\_expression2 ] )

**ln**

Renvoie le logarithme naturel de "numeric\_expression1".

**Syntaxe**

ln ( numeric\_expression )

**cbrt**

Renvoie la racine cubique de "numeric\_expression1".

**Syntaxe**

cbrt ( numeric\_expression )

**div**

Renvoie le quotient entier de l'expression numérique "numeric\_expression1" divisée par "numeric\_expression2".

**Syntaxe**

div ( numeric\_expression1 , numeric\_expression2 )

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

pi ( )

## Fonctions trigonométriques Postgres

### **acos**

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

### **Syntaxe**

```
acos ( numeric_expression )
```

### **asin**

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

### **Syntaxe**

```
asin ( numeric_expression )
```

### **atan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

### **Syntaxe**

```
atan ( numeric_expression )
```

### **atan2**

Renvoie l'arc tangente des coordonnées x et y définies respectivement par "numeric\_expression1" et "numeric\_expression2", en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression2"/"numeric\_expression1".

### **Syntaxe**

```
atan2 ( numeric_expression1 , numeric_expression2 )
```

### **cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

### **cot**

Renvoie la cotangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cot ( numeric_expression )
```

### **degrees**

Renvoie une valeur en degrés, où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
degrees ( numeric_expression )
```

## **radians**

Renvoie une valeur en radians, où "numeric\_expression" représente un angle exprimé en degrés.

### **Syntaxe**

```
radians ( numeric_expression )
```

## **sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
sin ( numeric_expression )
```

## **tan**

Renvoie la tangente de "numeric\_expression" où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
tan ( numeric_expression )
```

## **ascii**

Renvoie un nombre correspondant au code ASCII du premier caractère de "string\_expression". Exemple : ascii('A') est 65.

### **Syntaxe**

```
ascii ( string_expression )
```

## **chr**

Renvoie le caractère qui correspond au code ASCII indiqué par "integer\_expression". La valeur de "integer\_expression" doit être comprise entre 0 et 255.

### **Syntaxe**

```
chr ( integer_expression )
```

## **{current\_db}**

### **Syntaxe**

```
current_database()
```

## **{current\_catalog}**

### **Syntaxe**

```
{current_catalog}
```

## **{current\_schema}**

### **Syntaxe**

```
{current_schema}
```

## **{current\_user}**

### **Syntaxe**

```
{current_user}
```

## **{session\_user}**

### **Syntaxe**

{session\_user}

## **translate**

Renvoie l'expression de chaîne "string\_expression1", dans laquelle chaque occurrence de caractère contenue dans "string\_expression2" est remplacée par le caractère correspondant de "string\_expression3".

### **Syntaxe**

```
translate ( string_expression1 , string_expression2 ,  
string_expression3 )
```

## **date\_trunc**

Renvoie l'horodatage selon la précision spécifiée.

### **Syntaxe**

```
date_trunc ( string_expression , timestamp_expression )
```

## **version**

Renvoie la valeur de chaîne de la version de base de données.

### **Syntaxe**

```
version ( )
```

---

## **Red Brick**

### **ceil**

Renvoie le plus petit entier supérieur ou égal à "numeric\_expression" ou "string\_expression". Notez que "string\_expression" doit représenter une valeur numérique valide.

### **Syntaxe**

```
ceil ( numeric_expression|string_expression )
```

### **concat**

Renvoie la chaîne résultant de la concaténation ou de la jointure de "string\_expression1" et "string\_expression2".

### **Syntaxe**

```
concat ( string_expression1 , string_expression2 )
```

### **Exemple**

```
concat ( [Sales (query)].[Sales staff].[First name], [Sales (query)].  
[Sales staff].[Last name] )
```

Résultat

Renvoie le prénom et le nom ; par exemple, Bob Smith.

## **{current\_user}**

Renvoie le nom (identificateur d'autorisation) de l'utilisateur en cours de la base de données.

### **Syntaxe**

```
{ current_user }
```

## **date**

Renvoie une valeur de date. "Expression" peut être constitué de caractères ou d'un horodatage.

### **Syntaxe**

```
date ( expression )
```

## **dateadd**

Ajoute un intervalle "interval" à "datetime\_expression" et renvoie un résultat qui est du même type date-heure que "datetime\_expression". L'expression "datepart" représente l'année, le mois, le jour, l'heure, les minutes et les secondes. La valeur de "interval" doit être un entier et l'expression "datetime\_expression" peut être une date, une heure ou un horodatage.

### **Syntaxe**

```
dateadd ( { datepart } , interval, datetime_expression )
```

## **datediff**

Détermine la différence entre deux expressions de date-heure et renvoie un résultat sous forme d'entier en unités de "datepart". L'expression "datepart" représente l'année, le mois, le jour, l'heure, les minutes ou les secondes. Les valeurs de "datetime\_expression1" et "datetime\_expression2" peuvent être des dates, des heures ou des horodatages.

### **Syntaxe**

```
datediff ( { datepart } , datetime_expression1, datetime_expression2 )
```

## **datetimeame**

Extrait "datepart" de "datetime\_expression" et renvoie sa valeur sous forme de chaîne de caractères. L'expression "datepart" représente l'année, le mois, le jour, l'heure, les minutes ou les secondes. "datetime\_expression" peut être une date, une heure ou un horodatage.

### **Syntaxe**

```
datetimeame ( { datepart } , datetime_expression )
```

## **dec**

Convertit une valeur "expression" en une valeur dont le type de données est décimal (précision, échelle). Le degré de précision par défaut est 9. La valeur d'échelle par défaut est 0.

### **Syntaxe**

```
dec ( expression , [ precision , scale ] )
```

## decimal

Convertit une valeur "expression" en une valeur dont le type de données est décimal (précision, échelle). Le degré de précision par défaut est 9. La valeur d'échelle par défaut est 0.

### Syntaxe

```
decimal ( expression , [ precision , scale ] )
```

## decode

Compare et convertit "expression" en une autre valeur. Si "expression" correspond à la cible "target", elle est remplacée ; sinon, elle est remplacée par la valeur par défaut "default" ou par NULL si aucune valeur par défaut n'a été définie. Les expressions peuvent être de n'importe quel type, mais elles doivent toutes être du même type.

### Syntaxe

```
decode ( expression , target , remplacement [ ,default ] )
```

## float

Convertit "numeric\_expression" en un nombre à virgule flottante et double précision.

### Syntaxe

```
float ( numeric_expression )
```

## ifnull

Trouve les valeurs manquantes de "expression" et les remplace par une valeur "substitute". Si "expression" a la valeur NULL, "substitute" est renvoyé, sinon la valeur de "expression" est renvoyée. Les expressions peuvent être de n'importe quel type, mais elles doivent toutes être du même type.

### Syntaxe

```
ifnull ( expression, substitute )
```

## int

Convertit "numeric\_expression" en valeur entière et la renvoie. Si "numeric\_expression" est NULL, renvoie NULL.

### Syntaxe

```
int ( numeric_expression )
```

## integer

Convertit "numeric\_expression" en valeur entière et la renvoie. Si "numeric\_expression" est NULL, renvoie NULL.

### Syntaxe

```
integer ( numeric_expression )
```

### Exemple

```
integer ( 84.95 )
```

Résultat



**length**

Renvoie un résultat sous forme d'entier précisant le nombre de caractères dans "string\_expression". Si "string\_expression" a une valeur Null, renvoie Null.

**Syntaxe**

```
length ( string_expression )
```

**lengthb**

Renvoie un résultat sous forme d'entier précisant le nombre d'octets dans "string\_expression". Si "expression\_chaine" est Null, renvoie Null.

**Syntaxe**

```
lengthb ( string_expression )
```

**ltrim**

Supprime les blancs de gauche de "string\_expression". Si "string\_expression" est Null, renvoie Null.

**Syntaxe**

```
ltrim ( string_expression )
```

**nullif**

Renvoie Null si "expression1" et "expression2" ont la même valeur. Si elles ont des valeurs différentes, la valeur de "expression1" est renvoyée. "Expression1" et "expression2" peuvent être de n'importe quel type, tant qu'elles sont du même type.

**Syntaxe**

```
nullif ( expression1 , expression2 )
```

**positionb**

Renvoie un entier qui dépend de la position de départ (numéro d'octet) de "string\_expression1" dans "string\_expression2". Si l'expression "string\_expression1" n'est pas trouvée, le résultat est 0. Si "string\_expression1" a une longueur nulle, le résultat est 1. Si "string\_expression1" est Null, un message d'erreur est renvoyé. Si "string\_expression2" est Null, la valeur renvoyée est 0.

**Syntaxe**

```
positionb ( string_expression1, string_expression2 )
```

**real**

Renvoie une valeur réelle. Si "numeric\_expression" est NULL, renvoie NULL.

**Syntaxe**

```
real ( numeric_expression )
```

**round**

Renvoie l'expression "numeric\_expression" arrondie à la valeur la plus proche avec "integer\_expression" positions à droite du séparateur décimal. Si

"integer\_expression" est une valeur négative, l'expression "numeric\_expression" est arrondie à la valeur absolue la plus proche avec "integer\_expression" positions à gauche du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression , integer_expression )
```

### Exemple

```
round (125, -1)
```

Résultat

130

## rtrim

Supprime les blancs de droite de "string\_expression". Si "string\_expression" est Null, renvoie Null.

### Syntaxe

```
rtrim ( string_expression )
```

### Exemple

```
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à la fin de ces derniers.

## sign

Détermine le signe de "numeric\_expression" et renvoie 1 pour une valeur positive, -1 pour une valeur négative et 0 pour zéro.

### Syntaxe

```
sign ( numeric_expression )
```

## string

Convertit "expression" en une chaîne de caractères. "Expression" peut avoir une valeur numérique ou de date-heure.

### Syntaxe

```
string ( expression [ , length [ , scale ] ] )
```

## substr

Renvoie une sous-chaîne "string\_expression" qui commence à la position "start\_integer" et continue sur une longueur de "length\_integer" caractères. Si "length\_integer" n'est pas précisé, une sous-chaîne de "start\_integer" jusqu'à la fin de "string\_expression" est renvoyée.

### Syntaxe

```
substr ( string_expression , start_integer , length_integer )
```

### Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

### substrb

Renvoie une sous-chaîne "string\_expression" qui commence à la position "start\_integer" et continue sur une longueur de "length\_integer" octets. Si "length\_integer" n'est pas précisé, une sous-chaîne de "start\_integer" jusqu'à la fin de "string\_expression" est renvoyée.

### Syntaxe

```
substrb ( string_expression , start_integer , length_integer )
```

### time

Crée une valeur d'heure à partir d'une "expression", qui peut être une chaîne de caractères ou une expression de type date-heure.

### Syntaxe

```
time ( expression )
```

### timestamp

Crée une valeur d'horodatage à partir de "timestamp\_expression" qui est une chaîne de caractères.

### Syntaxe

```
timestamp ( timestamp_expression )
```

### timestamp

Crée une valeur date-heure à partir de "time\_expression" et "date\_expression". Si "time\_expression" ou "date\_expression" est Null, l'expression de date-heure résultante sera également Null.

### Syntaxe

```
timestamp ( date_expression , time_expression )
```

### to\_char

Convertit "date\_source" en une chaîne de caractères définie par "format\_string". "source\_date" peut avoir un type de données date, heure ou horodatage.

### Syntaxe

```
to_char ( source_date, format_string )
```

---

## SAP BW

### Fonctions trigonométriques SAP BW

#### arccos

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

**Syntaxe**

`arccos ( numeric_expression )`

**arcsin**

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

**Syntaxe**

`arcsin ( numeric_expression )`

**arctan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

**Syntaxe**

`arctan ( numeric_expression )`

**cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`cos ( numeric_expression )`

**sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`sin ( numeric_expression )`

**tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`tan ( numeric_expression )`

**coshyp**

Renvoie le cosinus hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`coshyp ( numeric_expression )`

**sinhyp**

Renvoie le sinus hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`sinhyp ( numeric_expression )`

**tanhyp**

Renvoie la tangente hyperbolique de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

`tanhyp ( numeric_expression )`

## Fonctions mathématiques SAP BW

**log10**

Renvoie le logarithme à base 10 de "numeric\_expression".

**Syntaxe**

`log10 ( numeric_expression )`

---

## Salesforce.com

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Fonctions de date

**CALENDAR\_MONTH**

Renvoie un nombre représentant le mois calendaire de "expression\_date".

**Syntaxe**

`CALENDAR_MONTH ( date_expression )`

**Exemple**

`CALENDAR_MONTH ( '2012-02-29' )`

Résultat

2

**CALENDAR\_QUARTER**

Renvoie un nombre représentant le trimestre calendaire de "expression\_date".

**Syntaxe**

`CALENDAR_QUARTER ( date_expression )`

**Exemple**

`CALENDAR_QUARTER ( '2012-02-29' )`

Résultat

1

**CALENDAR\_YEAR**

Renvoie un nombre représentant l'année calendaire de "expression\_date".

**Syntaxe**

`CALENDAR_YEAR ( date_expression )`

**Exemple**

`CALENDAR_YEAR ( '2012-02-29' )`

Résultat

2012

### **DAY\_IN\_MONTH**

Renvoie un nombre représentant le jour du mois de "expression\_date".

#### **Syntaxe**

```
DAY_IN_MONTH ( date_expression )
```

#### **Exemple**

```
DAY_IN_MONTH ( '2012-02-29' )
```

Résultat

29

### **DAY\_IN\_WEEK**

Renvoie un nombre représentant le jour de la semaine de "expression\_date", 1 pour dimanche, 7 pour samedi.

#### **Syntaxe**

```
DAY_IN_WEEK ( date_expression )
```

#### **Exemple**

```
DAY_IN_WEEK ( '2012-02-29' )
```

Résultat

4 (mercredi)

### **DAY\_IN\_YEAR**

Renvoie un nombre représentant le jour de l'année dans "expression\_date".

#### **Syntaxe**

```
DAY_IN_YEAR ( date_expression )
```

#### **Exemple**

```
DAY_IN_YEAR ( '2012-02-29' )
```

Résultat

60

### **DAY\_ONLY**

Renvoie une date représentant la portion jour de "expression\_date-heure".

#### **Syntaxe**

```
DAY_ONLY ( dateTime_expression )
```

#### **Exemple**

```
DAY_ONLY ( '2012-02-29T23:00:01Z' )
```

Résultat

2012-02-29

## **FISCAL\_MONTH**

Renvoie un nombre représentant le mois fiscal de "expression\_date". Le résultat peut être différent de celui de CALENDAR\_MONTH() si l'exercice fiscal de votre organisation ne correspond pas à l'année du calendrier grégorien. Si votre exercice fiscal débute en mars, 1 pour mars, 12 pour février.

### **Syntaxe**

```
FISCAL_MONTH ( date_expression )
```

### **Exemple**

```
FISCAL_MONTH ( '2012-02-29' )
```

Résultat

12

## **FISCAL\_QUARTER**

Renvoie un nombre représentant le trimestre fiscal de "expression\_date". Le résultat peut être différent de celui de CALENDAR\_QUARTER() si l'exercice fiscal de votre organisation ne correspond pas à l'année du calendrier grégorien. Si votre exercice fiscal débute en juillet, 1 pour le 15 juillet, 4 pour le 6 juin.

### **Syntaxe**

```
FISCAL_QUARTER ( date_expression )
```

### **Exemple**

```
FISCAL_QUARTER ( '2012-02-29' )
```

Résultat

3

## **FISCAL\_YEAR**

Renvoie un nombre représentant l'exercice fiscal de "expression\_date". Le résultat peut être différent de celui de CALENDAR\_YEAR() si l'exercice fiscal de votre organisation ne correspond pas à l'année du calendrier grégorien.

### **Syntaxe**

```
FISCAL_YEAR ( date_expression )
```

### **Exemple**

```
FISCAL_YEAR ( '2012-02-29' )
```

Résultat

2012

## **HOUR\_IN\_DAY**

Renvoie un nombre représentant l'heure du jour de "expression\_date-heure".

### **Syntaxe**

```
HOUR_IN_DAY ( dateTime_expression )
```

**Exemple**

```
HOUR_IN_DAY ( '2012-02-29T23:00:01Z' )
```

Résultat

23

**WEEK\_IN\_MONTH**

Renvoie un nombre représentant la semaine du mois de "expression\_date". La première semaine commence le premier jour du mois et se termine le septième.

**Syntaxe**

```
WEEK_IN_MONTH ( date_expression )
```

**Exemple**

```
WEEK_IN_MONTH ( '2012-02-29' )
```

Résultat

5

**WEEK\_IN\_YEAR**

Renvoie un nombre représentant la semaine calendaire de "expression\_date". La première semaine commence le premier janvier et se termine le sept.

**Syntaxe**

```
WEEK_IN_YEAR ( date_expression )
```

**Exemple**

```
WEEK_IN_YEAR ( '2012-02-29' )
```

Résultat

9

**convertCurrency**

Convertit "expression\_numérique" en devise de l'utilisateur lorsque le support multidevise est activé.

**Syntaxe**

```
convertCurrency ( numeric_expression )
```

**convertTimezone**

Convertit "expression\_date-heure" vers le fuseau horaire de l'utilisateur.

**Syntaxe**

```
convertTimezone ( dateTime_expression )
```



### Fonctions mathématiques SQL Server

#### **log**

Renvoie le logarithme naturel de "numeric\_expression".

#### **Syntaxe**

```
log ( numeric_expression )
```

#### **log10**

Renvoie le logarithme à base 10 de "numeric\_expression".

#### **Syntaxe**

```
log10 ( numeric_expression )
```

#### **pi**

Renvoie un nombre en virgule flottante représentant la valeur de la constante pi.

#### **Syntaxe**

```
pi ( )
```

#### **rand**

Renvoie un nombre aléatoire produit en utilisant "integer\_expression" comme valeur de départ.

#### **Syntaxe**

```
rand ( integer_expression )
```

### Fonctions trigonométriques SQL Server

#### **acos**

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

#### **Syntaxe**

```
acos ( numeric_expression )
```

#### **asin**

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

#### **Syntaxe**

```
asin ( numeric_expression )
```

#### **atan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

#### **Syntaxe**

```
atan ( numeric_expression )
```

## **atn2**

Renvoie l'arc tangente des coordonnées x et y définies respectivement par "numeric\_expression1" et "numeric\_expression2", en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression1".

### **Syntaxe**

```
atn2 ( numeric_expression1, numeric_expression2 )
```

## **cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

## **cot**

Renvoie la cotangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cot ( numeric_expression )
```

## **degrées**

Renvoie l'expression "numeric\_expression" convertie en degrés.

### **Syntaxe**

```
degrées ( numeric_expression )
```

## **radians**

Convertit "numeric\_expression" degrés en radians.

### **Syntaxe**

```
radians ( numeric_expression )
```

## **sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
sin ( numeric_expression )
```

## **tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
tan ( numeric_expression )
```

## **ascii**

Renvoie un nombre correspondant au code ASCII du premier caractère de "string\_expression".

### **Syntaxe**

```
ascii ( string_expression )
```

### Exemple

```
ascii ( 'A' )
```

Résultat

65

## char

Renvoie le caractère qui correspond au code ASCII indiqué par "integer\_expression". La valeur de "integer\_expression" doit être comprise entre 0 et 255.

### Syntaxe

```
char ( integer_expression )
```

### Exemple

```
char ( 65 )
```

Résultat

A

## charindex

Recherche la première occurrence de "string\_expression1" dans "string\_expression2" et renvoie un entier. "start\_location" correspond à la position du caractère où commence la recherche de "string\_expression1" dans "string\_expression2". Si "start\_location" n'est pas indiqué, s'il s'agit d'un nombre négatif ou d'un zéro, la recherche commence au début de "string\_expression2".

### Syntaxe

```
charindex ( string_expression1 , string_expression2 [ ,  
start_location ] )
```

## {current\_user}

Renvoie le nom de l'utilisateur en cours.

### Syntaxe

```
{ current_user }
```

## datalength

Renvoie la longueur en octets de "string\_expression".

### Syntaxe

```
datalength ( string_expression )
```

## dateadd

Renvoie la date résultant de l'ajout de "integer\_expression" unités (indiquées par "datepart" (jour, mois, année)) à "date\_expression".

### Syntaxe

```
dateadd ( { datepart } , integer_expression , date_expression )
```

## datediff

Renvoie le nombre d'unités "datepart" (jour, mois, année) entre "date\_expression1" et "date\_expression2".

### Syntaxe

```
datediff ( {datepart} , date_expression1 , date_expression2 )
```

### Exemple

```
datediff ( {yy} , 1984-01-01 , 1997-01-01 )
```

Résultat

13

## datetime

Renvoie "datepart" à partir de "date\_expression", qui peut être de type date-heure, smalldatetime ou heure sous forme de chaîne ASCII. Notez que "datepart" doit être un mot clé représentant une partie de date, ou son abréviation reconnue par Microsoft® SQL Server, et doit apparaître entre accolades.

### Syntaxe

```
datetime ( ' { ' datepart ' } ' , date_expression )
```

### Exemple

```
datetime ( {mm} , 2000-01-01 )
```

Résultat

Janvier

## datepart

Renvoie une partie de "date\_expression" (par exemple le mois) sous forme d'un entier. La valeur de "date\_expression" peut être de type date-heure, smalldatetime, date, ou heure. Notez que "datepart" doit être un mot clé représentant une partie de date, ou son abréviation reconnue par Microsoft® SQL Server, et doit apparaître entre accolades.

### Syntaxe

```
datepart ( ' { ' datepart ' } ' , date_expression )
```

### Exemple

```
datepart ( {wk} , 2000-01-01 )
```

Résultat

1 (première semaine de l'année)

## day

Renvoie la partie jour de "date\_expression". Même résultat que extract (jour de date\_expression).

### Syntaxe

```
day ( date_expression )
```

## difference

Renvoie l'entier représentant la différence entre les valeurs de la fonction SOUNDEX spécifique de source de données pour les chaînes "string\_expression1" et "string\_expression2". La valeur renvoyée est comprise entre 0 et 4, 4 étant le meilleur résultat. Une valeur de 4 ne signifie pas que les chaînes sont égales.

### Syntaxe

```
difference ( string_expression1 , string_expression2 )
```

### Exemple 1

```
difference ([Sales target (query)].[Sales Staff].[First name],[Sales (query)].[Retailers].[Contact first name])
```

Résultat

0

### Exemple 2

```
difference ([Sales target (query)].[Sales Staff].[First name],[Sales target (query)].[Sales Staff].[First name])
```

Résultat

4

## getdate

Renvoie une valeur de date-heure représentant la date et l'heure en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

### Syntaxe

```
getdate ()
```

## isnull

Renvoie le premier argument non nul (ou la valeur NULL si les deux arguments sont nuls). Il est recommandé d'utiliser la fonction COALESCE standard sql à la place.

### Syntaxe

```
isnull ( expression , expression )
```

### Exemple

```
isnull ( [Sales (query)].[Sales Fact].[Sales quantity] , 0 )
```

Résultat

Renvoie la quantité des ventes si elle n'est pas nulle, sinon renvoie 0.

## left

Renvoie les "integer\_expression" premiers caractères de "string\_expression".

### Syntaxe

```
left ( string_expression , integer_expression )
```

### **Exemple**

```
left ( [Sales (query)].[Sales staff].[Last name] , 3 )
```

Résultat

Renvoie les trois premiers caractères de chaque nom.

### **ltrim**

Renvoie "string\_expression" en supprimant les espaces à gauche.

### **Syntaxe**

```
ltrim ( string_expression )
```

### **month**

Renvoie la partie mois de "date\_expression". Même résultat que extract (mois de date\_expression).

### **Syntaxe**

```
month ( date_expression )
```

### **patindex**

Renvoie un entier qui représente la position de début de la première occurrence de "string\_expression1" dans "string\_expression2". Renvoie 0 si "string\_expression1" est introuvable. Le caractère générique % doit précéder et suivre "string\_expression1", sauf lorsque vous recherchez les premiers et derniers caractères.

### **Syntaxe**

```
patindex ( string_expression1 , string_expression2 )
```

### **Exemple**

```
patindex ( '%po%', 'Report' )
```

Résultat

3

### **replace**

Remplace toutes les occurrences de "string\_expression2" par "string\_expression3" dans "string\_expression1".

### **Syntaxe**

```
replace ( string_expression1 , string_expression2 ,  
string_expression3 )
```

### **replicate**

Renvoie une chaîne résultant de la répétition de "string\_expression" un nombre de fois déterminé par "integer\_expression".

### **Syntaxe**

```
replicate ( string_expression , integer_expression )
```

## inverse

Renvoie "string\_expression" en ordre inversé.

### Syntaxe

```
reverse ( string_expression )
```

## right

Renvoie les "integer\_expression" derniers caractères de "string\_expression".

### Syntaxe

```
right ( string_expression , integer_expression )
```

## round

Renvoie l'expression "numeric\_expression" arrondie à la valeur la plus proche avec "integer\_expression" positions à droite du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression , integer_expression )
```

### Exemple

```
round (125, -1)
```

Résultat

130

## rtrim

Renvoie "string\_expression" en supprimant les espaces de fin.

### Syntaxe

```
rtrim ( string_expression )
```

### Exemple

```
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à la fin de ces derniers.

## sign

Renvoie +1 si l'expression "numeric\_expression" est positive, 0 si elle est égale à zéro ou -1 si elle est négative.

### Syntaxe

```
sign ( numeric_expression )
```

## soundex

Renvoie une chaîne de quatre caractères constituant une représentation phonétique de "string\_expression".

## Syntaxe

soundex ( string\_expression )

## space

Renvoie une chaîne formée de "integer\_expression" espaces.

## Syntaxe

space ( integer\_expression )

## str

Renvoie une représentation sous forme de chaîne de "numeric\_expression". La valeur de "integer\_expression1" correspond à la longueur de la chaîne renvoyée. La valeur de "integer\_expression2" correspond au nombre de décimales.

## Syntaxe

str ( numeric\_expression [ , integer\_expression1 [ , integer\_expression2 ] ] )

## stuff

Renvoie une chaîne qui résulte de la suppression dans "string\_expression1" de "integer\_expression2" caractères à partir de la position "integer\_expression1", puis de l'insertion de "string\_expression2" au début de "string\_expression1". Le premier caractère d'une chaîne se trouve en position 1.

## Syntaxe

stuff ( string\_expression1 , integer\_expression1 , integer\_expression2 , string\_expression2 )

## year

Renvoie la partie année de "date\_expression". Même résultat que extract (année de date\_expression).

## Syntaxe

year ( date\_expression )

---

## Sybase

### Fonctions mathématiques Sybase

#### log

Renvoie le logarithme naturel de "numeric\_expression".

#### Syntaxe

log ( numeric\_expression )

#### log10

Renvoie le logarithme à base 10 de "numeric\_expression".

#### Syntaxe

log10 ( numeric\_expression )



**pi**

Renvoie un nombre en virgule flottante représentant la valeur de la constante pi.

**Syntaxe**

pi ( )

**sign**

Renvoie un indicateur correspondant au signe de l'expression "numeric\_expression" : +1 si l'expression est positive, 0 si elle est égale à zéro ou -1 si elle est négative.

**Syntaxe**

sign ( numeric\_expression )

## Fonctions trigonométriques Sybase

**acos**

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

**Syntaxe**

acos ( numeric\_expression )

**asin**

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

**Syntaxe**

asin ( numeric\_expression )

**atan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

**Syntaxe**

atan ( numeric\_expression )

**tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

tan ( numeric\_expression )

**atan2**

Renvoie l'angle (en radians) dont la tangente est "numeric\_expression1" / "numeric\_expression2".

**Syntaxe**

atan2 ( numeric\_expression1, numeric\_expression2 )

**cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
cos ( numeric_expression )
```

**cot**

Renvoie la cotangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
cot ( numeric_expression )
```

**degrees**

Convertit "numeric\_expression" radians en degrés.

**Syntaxe**

```
degrees ( numeric_expression )
```

**radians**

Renvoie l'équivalent (en degrés) de "numeric\_expression". Les résultats sont de même type que "numeric\_expression". Pour les expressions numériques ou décimales, les résultats ont une précision interne de 77 et une échelle égale à celle de "numeric\_expression". Lorsque le type de données monétaire est utilisé, la conversion interne en virgule flottante peut entraîner une diminution de la précision.

**Syntaxe**

```
radians ( numeric_expression )
```

**sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
sin ( numeric_expression )
```

**ascii**

Renvoie un nombre correspondant au code ASCII du premier caractère de "string\_expression".

**Syntaxe**

```
ascii ( string_expression )
```

**Exemple**

```
ascii ( 'A' )
```

Résultat

65

**char**

Convertit "integer\_expression" en une valeur de caractère. Char est généralement utilisé comme inverse d'ascii quand l'expression "integer\_expression" doit être comprise entre 0 et 255. Si la valeur résultante est le premier octet d'un caractère à plusieurs octets, le caractère ne sera peut-être pas défini.

## Syntaxe

```
char ( integer_expression )
```

## charindex

Renvoie un entier représentant la position initiale de "string\_expression1" dans "string\_expression2". Si "string\_expression1" est introuvable, zéro est renvoyé. Si "string\_expression1" contient des caractères génériques, charindex traite ces derniers comme des valeurs littérales.

## Syntaxe

```
charindex ( string_expression1 , string_expression2 )
```

## datalength

Renvoie la longueur en octets de "string\_expression".

## Syntaxe

```
datalength ( string_expression )
```

## dateadd

Renvoie la date résultant de l'ajout de "integer\_expression" unités indiquées par "datepart" (jour, mois, année) à "date\_expression". Notez que l'argument "datepart" doit être placé entre accolades.

## Syntaxe

```
dateadd ( ' { ' datepart ' } ' , integer_expression ,  
date_expression )
```

## Exemple

```
dateadd ( {dd} , 16 , 1997-06-16 )
```

Résultat

2 juillet 1997

## datediff

Renvoie le nombre d'unités indiquées par "datepart" (jour, mois, année) entre "date\_expression1" et "date\_expression2". Notez que l'argument "datepart" doit être placé entre accolades.

## Syntaxe

```
datediff ( ' { ' datepart ' } ' , date_expression1 ,  
date_expression2 )
```

## Exemple

```
datediff ( {yy} , 1984-01-01 , 1997-01-01 )
```

Résultat

13

## datetime

Renvoie la partie\_date "datepart" de "date\_expression" sous forme de chaîne ASCII. La valeur de "date\_expression" peut être de type date-heure, smalldatetime, date,

ou heure. Notez que l'argument "datepart" doit être placé entre accolades. Notez que l'argument "partie\_date" doit être entre accolades.

### Syntaxe

```
datetime ( ' { ' datepart ' } ' , date_expression )
```

### Exemple

```
datetime ( {mm} , 1999-05-01 )
```

Résultat

Mai

## datepart

Renvoie la partie\_date "datepart" de "date\_expression" sous forme d'entier. La valeur de "date\_expression" peut être de type date-heure, smalldatetime, date, ou heure. Notez que l'argument "datepart" doit être placé entre accolades.

### Syntaxe

```
datepart ( ' { ' datepart ' } ' , date_expression )
```

### Exemple

```
datepart ( {mm} , 1999-05-01 )
```

Résultat

5

## jour

Renvoie le jour du mois (de 1 à 31) de "date\_expression".

### Syntaxe

```
day ( date_expression )
```

## difference

Renvoie l'entier représentant la différence entre les valeurs de la fonction SOUNDEX spécifique de source de données pour les chaînes "string\_expression1" et "string\_expression2". La valeur renvoyée se situe entre 0 et 4, où 4 représente le meilleur résultat. Une valeur de 4 ne signifie pas que les chaînes sont égales.

### Syntaxe

```
difference ( string_expression1 , string_expression2 )
```

### Exemple 1

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales (query)].[Retailers].[Contact first name])
```

Résultat

0

## Exemple 2

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])
```

Résultat

4

## getdate

Renvoie la date et l'heure système en cours.

### Syntaxe

```
getdate ( )
```

## left

Renvoie les "integer\_expression" premiers caractères de "string\_expression".

### Syntaxe

```
left ( string_expression , integer_expression )
```

### Exemple

```
left ( [Sales (query)].[Sales staff].[Last name] , 3 )
```

Résultat

Renvoie les trois premiers caractères de chaque nom.

## ltrim

Renvoie "string\_expression" en supprimant les blancs de début.

### Syntaxe

```
ltrim ( string_expression )
```

## month

Renvoie le numéro du mois (1 à 12) à partir de "date\_expression".

### Syntaxe

```
month ( date_expression )
```

## patindex

Renvoie un entier représentant la position initiale de la première occurrence de "string\_expression1" dans "string\_expression2", ou 0 lorsque "string\_expression1" est introuvable. Par défaut, patindex renvoie le décalage en termes de caractères. Le décalage peut être renvoyé en octets en définissant le type de renvoi sur les octets. Le caractère générique % doit précéder et suivre le motif dans "string\_expression1", sauf lorsque vous recherchez les premiers et derniers caractères.

### Syntaxe

```
patindex ( string_expression1 , string_expression2 [ using {bytes|chars|characters} ] )
```

## rand

Renvoie une valeur aléatoire à virgule flottante se situant entre 0 et 1, en utilisant "integer\_expression" comme valeur de départ.

### Syntaxe

```
rand ( integer_expression )
```

## replicate

Renvoie une chaîne résultant de la répétition de "string\_expression" un nombre de fois déterminé par "integer\_expression".

### Syntaxe

```
replicate ( string_expression , integer_expression )
```

## inverse

Renvoie "string\_expression" en ordre inversé.

### Syntaxe

```
reverse ( string_expression )
```

## right

Renvoie les "integer\_expression" derniers caractères de "string\_expression".

### Syntaxe

```
right ( string_expression , integer_expression )
```

## round

Renvoie l'expression "numeric\_expression" arrondie à la valeur la plus proche avec "integer\_expression" positions à droite du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression, integer_expression )
```

### Exemple

```
round (125, -1)
```

Résultat

130

## rtrim

Renvoie "string\_expression" en supprimant les espaces de fin.

### Syntaxe

```
rtrim ( string_expression )
```

### Exemple

```
rtrim ( [Sales (query)].[Sales staff].[Last name] )
```

Résultat

Renvoie les noms de famille après avoir supprimé les espaces à la fin de ces derniers.

## soundex

Renvoie un code "soundex" à 4 caractères pour les chaînes de caractères composées d'une séquence contiguë de lettres romaines valides à un ou à deux octets.

### Syntaxe

```
soundex ( string_expression )
```

## space

Renvoie un chaîne avec un espacement à un octet "integer\_expression".

### Syntaxe

```
space ( integer_expression )
```

## str

Renvoie une représentation sous forme de chaîne de "numeric\_expression". "Integer\_expression1" correspond à la longueur de la chaîne retournée et utilise une valeur par défaut égale à 10. "Integer\_expression2" correspond au nombre de décimales et utilise une valeur par défaut égale à 0. Ces deux valeurs sont facultatives.

### Syntaxe

```
str ( numeric_expression [ , integer_expression1 [ ,  
integer_expression2 ] ] )
```

## stuff

Supprime "integer\_expression2" caractères de "string\_expression1" en commençant à "integer\_expression1", puis insère "string\_expression2" dans "string\_expression1" à cette position. Pour supprimer les caractères sans insérer d'autres caractères, "string\_expression2" doit avoir la valeur Null et non " ", qui indique un espace simple.

### Syntaxe

```
stuff ( string_expression1 , integer_expression1 ,  
integer_expression2 , string_expression2 )
```

## substring

Renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1". "Integer\_expression2" indique le nombre de caractères dans la sous-chaîne.

### Syntaxe

```
substring ( string_expression , integer_expression1 ,  
integer_expression2 )
```

### Exemple

```
substring ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## to\_unichar

Renvoie une expression unichar ayant la valeur "integer\_expression". Si "integer\_expression" se situe dans la plage 0xD800..0xDFFF, l'opération est abandonnée. Si "integer\_expression" est dans la plage 0..0xFFFF, une valeur Unicode unique est renvoyée. Si "integer\_expression" se situe dans la plage 0x10000..0x10FFFF, une paire de substitution est renvoyée.

### Syntaxe

```
to_unichar ( integer_expression )
```

## uhighsurr

Renvoie 1 si la valeur Unicode à la position "integer\_expression" correspond à la valeur supérieure de la paire de substitution (qui devrait figurer en premier dans la paire). Sinon, la valeur 0 est renvoyée. Cette fonction vous permet d'écrire du code explicite pour la gestion des valeurs de substitution. Plus particulièrement, si une sous-chaîne commence avec un caractère Unicode dans lequel la fonction uhighsurr() est définie sur true, le système extrait une sous-chaîne d'au moins 2 valeurs Unicode, puisque la fonction substr() ne peut en extraire seulement 1. La fonction substr() n'extrait pas la moitié d'une paire de substitution.

### Syntaxe

```
uhighsurr ( string_expression , integer_expression )
```

## ulowsurr

Renvoie 1 si la valeur Unicode à la position "integer\_expression" correspond à la valeur inférieure de la paire de substitution (qui devrait figurer en deuxième dans la paire). Sinon, la valeur 0 est renvoyée. Cette fonction vous permet d'écrire du code explicite en fonction des ajustements apportés par les fonctions substr(), stuff() et right(). Plus particulièrement si une sous-chaîne se termine par une valeur Unicode dans laquelle la fonction ulowsurr() a la valeur true, le système extrait une sous-chaîne de 1 caractère de moins (ou de plus), puisque la fonction substr() n'extrait pas les chaînes qui contiennent une paire de substitution dans laquelle il manque un élément.

### Syntaxe

```
ulowsurr ( string_expression , integer_expression )
```

## uscalar

Renvoie la valeur Unicode scalaire du premier caractère Unicode de "string\_expression". Si le premier caractère ne correspond pas à la valeur supérieure d'une paire de substitution, la valeur se situe donc dans la plage 0..0xFFFF. Si le premier caractère correspond à la valeur supérieure d'une paire de substitution, la deuxième valeur doit être la valeur inférieure, et la valeur renvoyée doit se situer dans la plage 0x10000..0x10FFFF. Si la fonction est appelée pour un caractère Unicode contenant une paire de substitution dans laquelle il manque un élément, l'opération est abandonnée.

### Syntaxe

```
uscalar ( string_expression )
```



## year

Renvoie l'année à partir de "date\_expression".

### Syntaxe

```
year ( date_expression )
```

---

## Teradata

### Fonctions trigonométriques Teradata

#### acos

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression". La valeur de "numeric\_expression" doit être comprise entre -1 et 1 inclus.

#### Syntaxe

```
acos ( numeric_expression )
```

#### acosh

Renvoie le cosinus hyperbolique inverse de "numeric\_expression" où "numeric\_expression" peut être n'importe quel nombre réel égal ou supérieur à 1.

#### Syntaxe

```
acosh ( numeric_expression )
```

#### asin

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression". La valeur de "numeric\_expression" doit être comprise entre -1 et 1 inclus.

#### Syntaxe

```
asin ( numeric_expression )
```

#### asinh

Renvoie le sinus hyperbolique inverse de "numeric\_expression", où "numeric\_expression" peut être n'importe quel nombre réel.

#### Syntaxe

```
asinh ( numeric_expression )
```

#### atan

Renvoie l'arc tangente de "numeric\_expression" en radians, où l'arc tangent est l'angle dont la tangente a pour valeur "numeric\_expression".

#### Syntaxe

```
atan ( numeric_expression )
```

#### atan2

Renvoie l'arc tangente des coordonnées x et y définies respectivement par "numeric\_expression1" et "numeric\_expression2", en radians. L'angle renvoyé sera compris entre - et  $\pi$  radians,  $\pi$  non compris.

#### Syntaxe

```
atan2 ( numeric_expression1, numeric_expression2 )
```

**atanh**

Renvoie la tangente hyperbolique inverse de "numeric\_expression", où "numeric\_expression" peut être n'importe quel nombre réel entre 1 et -1 non compris.

**Syntaxe**

```
atanh ( numeric_expression )
```

**cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
cos ( numeric_expression )
```

**cosh**

Renvoie le cosinus hyperbolique de "numeric\_expression", où "numeric\_expression" peut être n'importe quel nombre réel.

**Syntaxe**

```
cosh ( numeric_expression )
```

**sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
sin ( numeric_expression )
```

**sinh**

Renvoie le sinus hyperbolique de "numeric\_expression", où "numeric\_expression" peut être n'importe quel nombre réel.

**Syntaxe**

```
sinh ( numeric_expression )
```

**tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
tan ( numeric_expression )
```

**tanh**

Renvoie la tangente hyperbolique de "numeric\_expression", où "numeric\_expression" peut être n'importe quel nombre réel.

**Syntaxe**

```
tanh ( numeric_expression )
```

**account**

Renvoie la chaîne de compte de l'utilisateur en cours.

## Syntaxe

{account}

## add\_months

Renvoie la date ou la date-heure résultant de l'addition de "integer\_expression" mois avec "date\_expression" ou "datetime\_expression".

## Syntaxe

```
add_months ( date_expression|datetime_expression ,
            integer_expression )
```

## bytes

Renvoie le nombre d'octets contenus dans "byte\_expression". "byte\_expression" est limité à BYTE ou VARBYTE.

## Syntaxe

```
bytes ( byte_expression )
```

## case\_n

Evalue "condition\_expression\_list" et renvoie la position de la première condition qui est évaluée à true, à condition qu'aucune des conditions précédentes dans la liste ne soit inconnue. Les mots clés doivent figurer entre accolades. La condition "no case" est une condition facultative qui est évaluée à true si toutes les expressions contenues dans la liste sont évaluées à false. La condition "no case" ou "unknown" est une condition facultative évaluée à true si chaque expression contenue dans la liste est évaluée à false, ou si une expression est évaluée à unknown et que toutes les conditions précédentes contenues dans la liste sont évaluées à false. La condition Unknown est une condition facultative qui est évaluée à true si une expression est évaluée à unknown et que toutes les conditions précédentes dans la liste sont évaluées à false.

## Syntaxe

```
case_n ( condition_expression_list [ , NO CASE|UNKNOWN|NO CASE OR
UNKNOWN [ , UNKNOWN ] ] )
```

## char2hexint

Renvoie la représentation hexadécimale de "string\_expression".

## Syntaxe

```
char2hexint ( string_expression )
```

## characters

Renvoie une valeur d'entier représentant le nombre de caractères logiques ou d'octets contenus dans "string\_expression".

## Syntaxe

```
characters ( string_expression )
```

## database

Renvoie le nom de la base de données par défaut de l'utilisateur en cours.

## Syntaxe

{database}

## date

Renvoie la date courante.

## Syntaxe

{date}

## format

Renvoie le format déclaré de "expression" sous forme de chaîne de caractères variable contenant jusqu'à 30 caractères.

## Syntaxe

format ( expression )

## index

Renvoie la position initiale de "string\_expression2" dans "string\_expression1".

## Syntaxe

index ( string\_expression1 , string\_expression2 )

## log

Calcule le logarithme à base 10 de "numeric\_expression". "numeric\_expression" doit être une expression numérique positive non nulle.

## Syntaxe

log ( numeric\_expression )

## nullif

Renvoie la valeur Null si les expressions "scalar\_expression1" et "scalar\_expression2" sont égales. Sinon, renvoie "scalar\_expression1". Les expressions "scalar\_expression1" et "scalar\_expression2" peuvent être de n'importe quel type.

## Syntaxe

nullif ( scalar\_expression1 , scalar\_expression2 )

## nullifzero

Si la valeur de "numeric\_expression" est égale à zéro, la fonction la convertit en valeur Null afin d'éviter la division par zéro.

## Syntaxe

nullifzero ( numeric\_expression )

## profile

Renvoie le profil en cours de la session ou la valeur Null s'il n'en existe aucun.

## Syntaxe

{profile}

## random

Renvoie un nombre entier aléatoire pour chaque ligne de la table des résultats. Les expressions "lower\_bound" et "upper\_bound" sont des constantes entières. Les limites pour les expressions "lower\_bound" et "upper\_bound" se situent entre -2147483648 et 2147483647, (inclus). La valeur de "upper\_bound" doit être supérieure ou égale à la valeur de "lower\_bound".

### Syntaxe

```
random ( lower_bound , upper_bound )
```

## role

Renvoie le rôle en cours de la session ou la valeur Null s'il n'en existe aucun.

### Syntaxe

```
{role}
```

## session

Renvoie le numéro de la session de l'utilisateur en cours.

### Syntaxe

```
{session}
```

## soundex

Renvoie une chaîne de caractères qui représente le code Soundex pour l'expression "string\_expression".

### Syntaxe

```
soundex ( string_expression )
```

## substr

Renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1" et dont la longueur est "integer\_expression2". Le premier caractère de "string\_expression" se trouve en position 1. Si l'expression "integer\_expression2" est omise, la fonction renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1" et se termine par la fin de "string\_expression".

### Syntaxe

```
substr ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

### Exemple

```
substr ( [Sales (query)].[Sales staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## time

Renvoie l'heure en cours basée sur le format 24 heures. Selon la documentation Teradata, la fonction système TIME est obsolète. Utilisez la fonction

CURRENT\_TIME à la place. Notez qu'en mode compatible, TIME renvoie un type de données heure, tandis qu'en mode de requête dynamique, TIME renvoie une valeur de type FLOAT.

### Syntaxe

{time}

## type

Renvoie le type de données défini pour "expression".

### Syntaxe

type ( expression )

## user

Renvoie le nom de l'utilisateur en cours.

### Syntaxe

{user}

## vargraphic

Renvoie une chaîne de caractères qui représente le code vargraphic pour l'expression "string\_expression".

### Syntaxe

vargraphic ( string\_expression )

## zeroifnull

Convertit les données Null en 0 pour éviter les erreurs créées par une valeur Null. Si "numeric\_expression" n'est pas Null, renvoie la valeur de "numeric\_expression". Si "numeric\_expression" est une chaîne de caractères, elle est convertie en valeur numérique de type de données Float. Si "numeric\_expression" est nulle ou égale à zéro, la fonction renvoie la valeur zéro.

### Syntaxe

zeroifnull ( numeric\_expression )

---

## Vectorwise

### Chaîne Vectorwise

#### lpad

Renvoie "string\_exp1" complétée à droite par "integer\_exp" occurrences de "string\_exp2". Si la longueur de "string\_exp1" dépasse "integer\_exp", la partie appropriée de "string\_exp1" est renvoyée.

#### Syntaxe

lpad ( string\_exp1, integer\_exp [ , string\_exp2 ] )

#### ltrim

Renvoie "string\_exp1", en supprimant les blancs de gauche.

**Syntaxe**

`ltrim ( string_exp1 )`

**rtrim**

Renvoie "string\_exp1", en supprimant les blancs de fin.

**Syntaxe**

`rtrim ( string_exp1 )`

**shift**

Renvoie "string\_exp1" décalé du nombre de caractères numeric\_exp. Si "numeric\_exp" est supérieur à zéro, le décalage est vers le droite, sinon il est vers la gauche.

**Syntaxe**

`shift ( string_exp1, numeric_exp )`

**soundex**

Renvoie un code à 4 caractères pour "string\_exp1".

**Syntaxe**

`soundex ( string_exp1 )`

**squeeze**

Renvoie "string\_exp1" en réduisant les séries de blancs à un blanc unique.

**Syntaxe**

`squeeze ( string_exp1 )`

## Fonctions mathématiques Vectorwise

**log**

Renvoie le logarithme en base 10 de "numeric\_exp1".

**Syntaxe**

`log ( numeric_exp1 )`

**ln**

Renvoie le logarithme naturel de "numeric\_exp1".

**Syntaxe**

`ln ( numeric_exp )`

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

`pi ( )`

## Fonctions trigonométriques Vectorwise

**acos**

Renvoie l'arc cosinus de "numeric\_exp" en radians. L'arc cosinus est l'angle dont le cosinus est "numeric\_exp".

**Syntaxe**

`acos ( numeric_exp )`

**asin**

Renvoie l'arc sinus de "numeric\_exp" en radians. L'arc sinus est l'angle dont le sinus est "numeric\_exp".

**Syntaxe**

`asin ( numeric_exp )`

**atan**

Renvoie l'arc tangente de "numeric\_exp" en radians. L'arc tangente est l'angle dont la tangente est "numeric\_exp".

**Syntaxe**

`atan ( numeric_exp )`

**atan2**

Renvoie l'arc tangente des coordonnées x et y spécifiées respectivement par "numeric\_exp1" et "numeric\_exp2", en radians. L'arc tangente est l'angle dont la tangente est "numeric\_exp2"/"numeric\_exp1".

**Syntaxe**

`atan2 ( numeric_exp1 ,numeric_exp2 )`

**cos**

Renvoie le cosinus de "numeric\_exp", où "numeric\_exp" représente un angle exprimé en radians.

**Syntaxe**

`cos ( numeric_exp )`

**sin**

Renvoie le sinus de "numeric\_exp", où "numeric\_exp" représente un angle exprimé en radians.

**Syntaxe**

`sin ( numeric_exp )`

**tan**

Renvoie la tangente de "numeric\_exp", où "numeric\_exp" correspond à un angle exprimé en radians.

**Syntaxe**

`tan ( numeric_exp )`

**{initial\_user}**

Renvoie le nom d'utilisateur initial.

**Syntaxe**

`{ initial_user }`

**{session\_user}**

Renvoie le nom d'utilisateur de session.



## Syntaxe

```
{ session_user }
```

## numeric\_trunc

Renvoie `expression_numérique1` tronquée à la position décimale, ou à `expression_numérique2` chiffres à gauche ou à droite de la position décimale.

## Syntaxe

```
trunc ( numeric_exp1 , numeric_exp2 )
```

---

## Vertica

### Chaîne Vertica

#### overlay

Renvoie "`string_expression1`" en remplaçant "`string_expression2`" avec la position de caractère "`numeric_expression`".

#### Syntaxe

```
overlay ( string_expression1 , string_expression2 ,  
numeric_expression1 [ , numeric_expression2 ] )
```

#### btrim

Renvoie `expression_chaine1` après avoir supprimé la chaîne de caractères la plus longue dans `expression_chaine2`.

#### Syntaxe

```
btrim ( string_expression1 [ , string_expression2 ] )
```

#### initcap

Renvoie l'expression "`string_expression`" mise en forme de la façon suivante : la première lettre de chaque mot est en majuscules et les autres lettres sont en minuscules. Les mots sont délimités par un blanc ou tout autre caractère non alphanumérique.

#### Syntaxe

```
initcap ( string_expression )
```

#### lpad

Renvoie l'expression "`string_expression1`" complétée par "`integer_expression`" occurrences de "`string_expression2`". Si la longueur de "`string_expression1`" dépasse "`integer_expression`", la partie appropriée de "`string_expression1`" est renvoyée.

#### Syntaxe

```
lpad ( string_expression1 , integer_expression [ , string_expression2  
] )
```

#### ltrim

Renvoie l'expression "`string_expression1`", dans laquelle les caractères de début sont supprimés jusqu'au premier caractère n'appartenant pas à "`string_expression2`". Exemple : `ltrim('xyxXxyAB', 'xy')` donne `'XxyAB'`.

#### Syntaxe

```
ltrim ( string_expression1 [ , string_expression2 ] )
```

### **to\_hex**

Renvoie une représentation sous forme de chaîne hexadécimale de "numeric\_expression1".

### **Syntaxe**

```
to_hex ( numeric_expression1 )
```

### **repeat**

Renvoie la chaîne "string\_expression" répétée "numeric\_expression1" fois.

### **Syntaxe**

```
repeat ( string_expression , numeric_expression1 )
```

### **replace**

Renvoie "string\_expression" en remplaçant "string\_expression2" par "string\_expression3".

### **Syntaxe**

```
replace ( string_expression , string_expression2 , string_expression3 )
```

### **rpad**

Renvoie l'expression "string\_expression1" complétée à droite par "integer\_expression" occurrences de "string\_expression2". Si la longueur de "string\_expression1" dépasse "integer\_expression", la partie appropriée de "string\_expression1" est renvoyée. Si "string\_expression2" n'est pas spécifiée, des espaces sont utilisés.

### **Syntaxe**

```
rpad ( string_expression1 , integer_expression [ , string_expression2 ] )
```

### **rtrim**

Renvoie la chaîne "string\_expression1" dans laquelle les derniers caractères sont supprimés après le dernier caractère qui n'appartient pas "string\_expression2".  
Exemple : rtrim ('ABxXxyx', 'xy') donne ABxX. Si "string\_expression2" n'est pas spécifiée, les caractères espace finaux sont supprimés.

### **Syntaxe**

```
rtrim ( string_expression1 [ , string_expression2 ] )
```

## **Formatage des types de données Vertica**

### **to\_char**

Renvoie la représentation sous forme de chaîne de "expression" au format "string\_expression". "expression" peut être une valeur de date ou une valeur numérique.

### **Syntaxe**

```
to_char ( expression , string_expression )
```

### **to\_date**

Convertit "string\_expression1" en une valeur de date de type "string\_expression2".

**Syntaxe**

`to_date ( string_expression1 , string_expression2 )`

**to\_number**

Convertit "string\_expression1" en une valeur numérique au format "string\_expression2".

**Syntaxe**

`to_number ( string_expression1, string_expression2 )`

**to\_timestamp**

Convertit "string\_expression1" en une valeur d'horodatage au format "string\_expression2".

**Syntaxe**

`to_timestamp ( string_expression1, string_expression2 )`

## Fonctions mathématiques Vertica

**log**

Renvoie le logarithme en base 10 de "numeric\_expression1" ou le logarithme en base "numeric\_expression2".

**Syntaxe**

`log ( numeric_expression1 [ , numeric_expression2 ] )`

**ln**

Renvoie le logarithme naturel de "numeric\_expression1".

**Syntaxe**

`ln ( numeric_expression )`

**cbrt**

Renvoie la racine cubique de "numeric\_expression1".

**Syntaxe**

`cbrt ( numeric_expression )`

**pi**

Renvoie la constante du nombre Pi.

**Syntaxe**

`pi ( )`

## Fonctions trigonométriques Vertica

**acos**

Renvoie l'arc cosinus de "numeric\_expression" exprimé en radians. L'arc cosinus est l'angle dont le cosinus a pour valeur "numeric\_expression".

**Syntaxe**

`acos ( numeric_expression )`

## **asin**

Renvoie l'arc sinus de "numeric\_expression" exprimé en radians. L'arc sinus est l'angle dont le sinus a pour valeur "numeric\_expression".

### **Syntaxe**

```
asin ( numeric_expression )
```

## **atan**

Renvoie l'arc tangente de "numeric\_expression" exprimé en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression".

### **Syntaxe**

```
atan ( numeric_expression )
```

## **atan2**

Renvoie l'arc tangente des coordonnées x et y définies respectivement par "numeric\_expression1" et "numeric\_expression2", en radians. L'arc tangente est l'angle dont la tangente a pour valeur "numeric\_expression2" / "numeric\_expression1".

### **Syntaxe**

```
atan2 ( numeric_expression1 , numeric_expression2 )
```

## **cos**

Renvoie le cosinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cos ( numeric_expression )
```

## **cot**

Renvoie la cotangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
cot ( numeric_expression )
```

## **degrees**

Renvoie une valeur en degrés, où "numeric\_expression" représente un angle exprimé en radians.

### **Syntaxe**

```
degrees ( numeric_expression )
```

## **radians**

Renvoie une valeur en radians, où "numeric\_expression" représente un angle exprimé en degrés.

### **Syntaxe**

```
radians ( numeric_expression )
```

## **sin**

Renvoie le sinus de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
sin ( numeric_expression )
```

**tan**

Renvoie la tangente de "numeric\_expression", où "numeric\_expression" représente un angle exprimé en radians.

**Syntaxe**

```
tan ( numeric_expression )
```

**ascii**

Renvoie un nombre correspondant au code ASCII du premier caractère de "expression\_chaine". Exemple : ascii('A') est 65.

**Syntaxe**

```
ascii ( string_expression )
```

**chr**

Renvoie le caractère qui correspond au code ASCII indiqué par "integer\_expression". La valeur de "integer\_expression" doit être comprise entre 0 et 255.

**Syntaxe**

```
chr ( integer_expression )
```

**current\_database**

Renvoie le nom de la base de données en cours.

**Syntaxe**

```
current_database ()
```

**current\_schema**

Renvoie le nom du schéma en cours.

**Syntaxe**

```
current_schema ()
```

**{current\_user}****Syntaxe**

```
{current_user}
```

**{session\_user}****Syntaxe**

```
{session_user}
```

**translate**

Renvoie l'expression "string\_expression1", dans laquelle chaque occurrence de caractère contenue dans "string\_expression2" est remplacée par le caractère correspondant de "string\_expression3".

## Syntaxe

```
translate ( string_expression1 , string_expression2 ,  
string_expression3 )
```

## date\_trunc

Renvoie l'horodatage selon la précision spécifiée.

## Syntaxe

```
date_trunc ( string_expression , timestamp_expression )
```

## version

Renvoie la valeur sous forme de chaîne de la version de base de données.

## Syntaxe

```
version ( )
```

---

## Fonctions macros

Cette liste répertorie les fonctions qui peuvent être utilisées dans une macro. Une macro peut contenir une ou plusieurs fonctions macro. Une macro débute et se termine par un signe dièse (#). Tout ce qui se trouve entre ces deux signes est considéré comme une expression de macro et est traité lors de l'exécution. Pour les fonctions macros qui acceptent des expressions de type horodatage avec fuseau horaire en tant qu'arguments, le format autorisé est 'yyyy-mm-dd hh:mm:ss[.ff]+hh:mm', dans lequel les fractions de secondes sont facultatives et peuvent être représentées par 1 à 9 chiffres. Le caractère 'T' " est également autorisé à la place de l'espace de séparation entre la partie date et la partie heure. En outre, au lieu du fuseau horaire '+hh:mm', le caractère 'Z' est accepté et il est traité en interne sous la forme '+00:00'. Les fonctions macros qui renvoient des expressions de type horodatage avec fuseau horaire renvoient 9 chiffres par défaut pour leurs fractions de secondes. Notez que la fonction macro timestampMask () peut être utilisée pour ajuster cette donnée en sortie, si besoin est.

+

Concatène deux chaînes.

## Syntaxe

```
value1 + value2
```

## Exemple

```
# '{ ' + $runLocale + ' }' #
```

Résultat

```
{en-us}
```

## \_add\_seconds

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" nombre de secondes à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

## Syntaxe

```
_add_seconds ( string_expression , integer_expression )
```

### Exemple 1

```
# _add_seconds ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

```
2005-11-01 11:59:59.000-05:00
```

### Exemple 2

```
# _add_seconds ( $current_timestamp , 1 ) #
```

Résultat

```
2005-11-01 12:00:01.000000000-05:00
```

### Exemple 3

```
# timestampMask ( _add_seconds ( $current_timestamp , 1 ) ,  
  'hh:mm:ss' ) #
```

Résultat

```
12:00:01
```

## \_add\_minutes

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" nombre de minutes à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

### Syntaxe

```
_add_minutes ( string_expression , integer_expression )
```

### Exemple 1

```
# _add_minutes ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

```
2005-11-01 11:59:00.000-05:00
```

### Exemple 2

```
# _add_minutes ( $current_timestamp , 1 ) #
```

Résultat

```
2005-11-01 12:01:00.000000000-05:00
```

### Exemple 3

```
# timestampMask ( _add_minutes ( $current_timestamp , 1 ) ,  
  'hh:mm:ss' ) #
```

Résultat

```
12:01:00
```

## \_add\_hours

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" nombre d'heures à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

### **Syntaxe**

```
_add_hours ( string_expression , integer_expression )
```

### **Exemple 1**

```
# _add_hours ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

```
2005-11-01 13:00:00.000-05:00
```

### **Exemple 2**

```
# _add_hours ( $current_timestamp , 1 ) #
```

Résultat

```
2005-11-01 13:00:00.000000000-05:00
```

### **Exemple 3**

```
# timestampMask ( _add_hours ( $current_timestamp , 1 ) ,  
'hh:mm:ss' ) #
```

Résultat

```
13:00:00
```

## \_add\_days

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" jours à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

### **Syntaxe**

```
_add_days ( string_expression , integer_expression )
```

### **Exemple 1**

```
# _add_days ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

```
2005-10-31 12:00:00.000000000-05:00
```

### **Exemple 2**

```
# _add_days ( $current_timestamp , 1 ) #
```

Résultat

```
2005-11-02 12:00:00.000000000-05:00
```



### Exemple 3

```
# timestampMask ( _add_days ( $current_timestamp , 1 ) ,  
'yyyy-mm-dd' ) #
```

Résultat

2005-11-02

## \_add\_months

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" mois à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

### Syntaxe

```
_add_months ( string_expression , integer_expression )
```

### Exemple 1

```
# _add_months ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

2005-10-01 12:00:00.000000000-05:00

### Exemple 2

```
# _add_months ( $current_timestamp , 1 ) #
```

Résultat

2005-12-01 12:00:00.000000000-05:00

### Exemple 3

```
# timestampMask ( _add_months ( $current_timestamp , 1 ) ,  
'yyyy-mm-dd' ) #
```

Résultat

2005-12-01

## \_add\_years

Renvoie l'horodatage avec fuseau horaire (sous forme de chaîne) résultant de l'ajout de "expression\_entier" années à "expression\_chaîne", où "expression\_chaîne" représente un horodatage avec fuseau horaire.

### Syntaxe

```
_add_years ( string_expression , integer_expression )
```

### Exemple 1

```
# _add_years ( '2005-11-01 12:00:00.000-05:00' , -1 ) #
```

Résultat

2004-11-01 12:00:00.000000000-05:00

### Exemple 2

```
# _add_years ( $current_timestamp , 1 ) #
```

Résultat

```
2006-11-01 12:00:00.000000000-05:00
```

### Exemple 3

```
# timestampMask ( _add_years ( $current_timestamp , 1 ) ,  
'yyyy-mm-dd' ) #
```

Résultat

```
2006-11-01
```

## array

Crée un tableau à partir de la liste des paramètres.

### Syntaxe

```
array ( string_expression|array_expression { ,  
string_expression|array_expression } )
```

### Exemple

```
# csv ( array ( 'a1' , array ( 'x1' , 'x2' ) , 'a2' ) ) #
```

Résultat

```
'a1' , 'x1' , 'x2' , 'a2'
```

## csv

Crée une chaîne à partir des éléments du tableau, dans laquelle les valeurs sont séparées par des virgules. Si vous le souhaitez, vous pouvez définir les chaînes de séparateurs et de guillemets à utiliser. Le séparateur par défaut est la virgule (,) et le caractère par défaut pour le guillemet est le guillemet simple (').

### Syntaxe

```
csv ( array_expression [ , separator_string [ , quote_string ] ] )
```

### Exemple

```
# csv ( array ( 'a1' , 'a2' ) ) #
```

Résultat

```
'a1' , 'a2'
```

## dq

Place "expression\_chaine" entre guillemets.

### Syntaxe

```
dq ( string_expression )
```

## Exemple

```
# dq ( 'zero' ) #
```

Résultat

```
"zéro"
```

## getConfigurationEntry

Extrait une entrée du fichier de configuration IBM® Cognos®. Le paramètre `force_decode_flag` est facultatif et doit prendre l'une des valeurs suivantes : 'true', '1', 1, 'false', '0', 0. La valeur par défaut est 'true'. Lorsque ce paramètre a pour valeur true, la valeur de l'entrée de configuration est déchiffrée sous forme de texte brut, si le chiffrement est utilisé.

## Syntaxe

```
getConfigurationEntry ( entry_string , force_decode_flag )
```

## Exemple

```
# getConfigurationEntry ( 'serverLocale' ) #
```

Résultat

```
fr
```

## grep

Recherche et renvoie les éléments d'un tableau qui correspondent au motif défini dans "chaîne\_motif".

## Syntaxe

```
grep ( pattern_string , array_expression )
```

## Exemple

```
# csv ( grep ( 's' , array ( 'as', 'an', 'arts' ) ) ) #
```

Résultat

```
'as', 'arts'
```

## \_first\_of\_month

Renvoie un horodatage avec fuseau horaire (sous forme de chaîne) convertissant la valeur du jour de "expression\_chaîne" à 1, où "expression\_chaîne" est un horodatage avec fuseau horaire.

## Syntaxe

```
_first_of_month ( string_expression )
```

## Exemple 1

```
# _first_of_month ( '2005-11-11 12:00:00.000-05:00' ) #
```

Résultat

```
2005-11-01 12:00:00.000000000-05:00
```

## Exemple 2

```
# timestampMask ( _first_of_month ( '2005-11-11 12:00:00.000-05:00' ) , 'yyymmdd' ) #
```

Résultat

20051101

## \_last\_of\_month

Renvoie un horodatage avec fuseau horaire (sous forme de chaîne) correspondant au dernier jour du mois représenté par "expression\_chaîne", où "expression\_chaîne" est un horodatage avec fuseau horaire.

### Syntaxe

```
_last_of_month ( string_expression )
```

### Exemple 1

```
# _last_of_month ( '2005-11-11 12:00:00.000-05:00' ) #
```

Résultat

2005-11-30 12:00:00.000000000-05:00

### Exemple 2

```
# timestampMask ( _last_of_month ( '2005-11-11 12:00:00.000-05:00' ) , 'yyyy-mm-dd' ) #
```

Résultat

2005-11-30

## join

Joint les éléments d'un tableau en utilisant "chaîne\_séparateur".

### Syntaxe

```
join ( separator_string , array_expression )
```

### Exemple

```
# sq ( join ( ' | | ' , array ( 'as', 'an', 'arts' ) ) ) #
```

Résultat

'as | | an | | arts'

## lstrip

Supprime les caractères de tête du premier argument. Le second argument facultatif définit le jeu de caractères à supprimer. Par défaut, cette fonction supprime les blancs (espaces, tabulations, retours chariot et sauts de ligne).

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
lstrip ( string_expression [ , set_of_characters ] )
```

### Exemple 1

```
# sql lstrip ( ' abc ' ) #
```

Résultat

```
'abc '
```

### Exemple 2

```
' ) #
```

Résultat

```
53,2100
```

## prompt

Invite l'utilisateur à fournir une valeur ou un membre unique. Seul l'élément "prompt\_name" est requis. Lorsque le type de données n'est pas spécifié, le système utilise le type par défaut 'string'. L'invite est facultative lorsque l'élément "defaultText" est spécifié. Lorsqu'il est indiqué, l'élément "text" précède la valeur. L'élément "queryItem" peut être défini afin de tirer parti des propriétés des informations d'invite de "queryItem". Lorsqu'il est indiqué, l'élément "trailing\_text" est ajouté à la valeur. Lorsque le type de données est 'memberuniquename', une invite de sélection de membre unique est créée. Il est recommandé d'entrer une valeur par défaut comme troisième argument. La valeur par défaut doit être un nom unique de membre valide. Elle peut aussi être un alias qui référence un nom unique de membre, comme dans Report Studio.

### Syntaxe

```
prompt ( prompt_name , datatype , defaultText , text , queryItem ,  
        trailing_text )
```

### Exemple 1

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > #prompt ( 'Starting CountryCode' , 'integer' , '10' ) #
```

Résultat

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > 10
```

### Exemple 2

```
[gosales].[COUNTRY].[COUNTRY] = # prompt ( 'countryPrompt' , 'string' , ''Canada'' ) #
```

Résultat

```
[gosales].[COUNTRY].[COUNTRY] = 'Canada'
```

### Exemple 3

```
set ( #prompt ('myProduct', 'memberuniquename', '[Camping Equipment]', '', '[Sales].[Product].[Product].[Product line]')#)
```

Résultat

```
set([Ventes].[Produit].[Produit].[Lignes de produits]->[all].[1])
```

## Remarques

- Le paramètre "defaultText" doit être spécifié de telle manière qu'il soit littéralement valide dans le contexte de la macro, car aucun formatage n'est effectué sur cette valeur. La chaîne par défaut "Canada" illustrée dans l'exemple 2 est spécifiée en tant que chaîne à l'aide de guillemets simples, dans laquelle ces derniers sont doublés, puis triplés. Par conséquent, la chaîne s'affiche correctement entre guillemets simples dans l'expression. En règle générale, lorsque l'élément "datatype" a pour valeur string, l'élément "defaultText" doit toujours être spécifié comme suit, sauf dans le contexte d'un paramètre de procédure stockée. Lorsque l'élément "defaultText" a pour type 'date' ou 'datetime', un format spécial doit être utilisé dans le contexte SQL. Voici des exemples de ces formats : 'DATE "2001-12-25"' et 'DATETIME "2001-12-25 12:00:00"'. Dans tous les autres contextes, utilisez le format date/datetime sans mot clé ni guillemets simples avec caractères d'échappement (par exemple, '2001-12-25').

## promptmany

Invite l'utilisateur à fournir une ou plusieurs valeurs ou un ou plusieurs membres. Seul l'élément "prompt\_name" est requis. Lorsque le type de données n'est pas spécifié, le système utilise le type par défaut 'string'. L'invite est facultative lorsque l'élément "defaultText" est spécifié. Lorsqu'il est indiqué, l'élément "text" précède la liste de valeurs. L'élément "queryItem" peut être défini afin de tirer parti des propriétés des informations d'invite de "queryItem". Lorsqu'il est indiqué, l'élément "trailing\_text" est ajouté à la liste de valeurs. Lorsque le type de données est 'memberuniquename', une invite de sélection multiple de membre est créée. Il est recommandé d'entrer une valeur par défaut comme troisième argument. La valeur par défaut doit être un nom unique de membre valide. Elle peut aussi être un alias qui référence un nom unique de membre, comme dans Report Studio.

## Syntaxe

```
promptmany ( prompt_name , datatype , defaultText , text , queryItem ,  
            trailing_text )
```

## Exemple 1

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY in ( # promptmany ( 'CountryName' ) # )
```

Résultat

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( 'Canada' , 'The Netherlands' , 'Russia' )
```

## Exemple 2

```
select . . . from gosales.gosales.dbo.COUNTRY_MULTILINGUAL  
COUNTRY_MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where  
COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE # promptmany ( 'Selected CountryCodes' , 'integer' , ' ' , 'and  
COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( ' , ' ' , ' ) ' ) #
```

Résultat

```
select . . . from gosales.gosales.dbo.COUNTRY_MULTILINGUAL  
COUNTRY_MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where  
COUNTRY_MULTILINGUAL.COUNTRY_CODE = XX.COUNTRY_CODE and  
COUNTRY_MULTILINGUAL.COUNTRY_CODE in ( 'Canada' , 'The  
Netherlands' , 'Russia' )
```

### Exemple 3

```
set ( #promptmany ('myProduct', 'memberuniqueName', '[Camping  
Equipment]', '', '[Sales].[Product].[Product].[Product line]')#)
```

Résultat

```
set([Ventes].[Produit].[Produit].[Lignes de produits]->[all].[1];  
[Ventes].[Produit].[Produit].[Lignes de produits]->[all].[2])
```

## rstrip

Supprime les caractères de fin du premier argument. Le second argument facultatif définit le jeu de caractères à supprimer. Par défaut, cette fonction supprime les blancs (espaces, tabulations, retours chariot et sauts de ligne).

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
rstrip ( string_expression [ , set_of_characters ] )
```

### Exemple 1

```
# sq( rstrip ( ' abc ' ) ) #
```

Résultat

```
' abc'
```

### Exemple 2

```
' ) #
```

Résultat

```
0053,21
```

## sb

Place "expression\_chaine" entre crochets.

### Syntaxe

```
sb ( string_expression )
```

### Exemple

```
# sb ( 'abc' ) #
```

Résultat

```
[abc]
```

## sq

Place "expression\_chaine" entre guillemets simples.

### Syntaxe

```
sq ( string_expression )
```

## Exemple

```
# sq ( 'zero' ) #
```

Résultat

```
'zéro'
```

## sort

Trie les éléments du tableau par ordre alphabétique. Les valeurs en double sont conservées.

## Syntaxe

```
sort ( array_expression )
```

## Exemple

```
# csv ( sort ( array ( 's3', 'a', 'x' ) ) ) #
```

Résultat

```
'a', 's3', 'x'
```

## split

Fractionne une chaîne ou les éléments d'une chaîne de tableau en éléments séparés.

## Syntaxe

```
split ( pattern_string, string_expression|array_expression )
```

## Exemple 1

```
# csv ( split ( '::', 'ab=c::de=f::gh=i' ) ) #
```

Résultat

```
'ab=c', 'de=f', 'gh=i'
```

## Exemple 2

```
# csv ( split ( '=', split ( '::', 'ab=c::de=f::gh=i' ) ) ) #
```

Résultat

```
'ab', 'c', 'de', 'f', 'gh', 'i'
```

## strip

Supprime les caractères de tête et de fin du premier argument. Le second argument facultatif définit le jeu de caractères à supprimer. Par défaut, cette fonction supprime les blancs (espaces, tabulations, retours chariot et sauts de ligne).

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

## Syntaxe

```
strip ( string_expression [ , set_of_characters ] )
```



### Exemple 1

```
# sq( strip ( ' abc ' ) ) #
```

Résultat

```
'abc'
```

### Exemple 2

```
' ) #
```

Résultat

```
53,21
```

## substitute

Recherche un motif dans une chaîne ou dans les éléments chaîne d'un tableau et remplace la première occurrence de "chaîne\_motif" par "chaîne\_replacement".

### Syntaxe

```
substitute ( pattern_string, replacement_string,  
string_expression|array_expression )
```

### Exemple 1

```
#sq ( substitute ( '^cn=', '***', 'cn=help' ) )#
```

Résultat

```
'***help'
```

### Exemple 2

```
# csv ( substitute ( '^cn=', '***', array ( 'cn=help' , 'acn=5' ) ) )  
#
```

Résultat

```
'***help', 'acn=5'
```

### Exemple 3

```
# csv ( substitute ( 'cn=', '', array ( 'cn=help' , 'acn=5' ) ) ) #
```

Résultat

```
'help', 'a5'
```

## timestampMask

Renvoie la chaîne "expression\_chaîne1", représentant un horodatage avec fuseau horaire, ajustée au format spécifié dans "expression\_chaîne2". La chaîne "expression\_chaîne2" doit être exprimée dans l'un des formats suivants : 'yyyy', 'mm', 'dd', 'yyyy-mm', 'yyyymm', 'yyyy-mm-dd', 'yyyymmdd', 'yyyy-mm-dd hh:mm:ss', 'yyyy-mm-dd hh:mm:ss+hh:mm', 'yyyy-mm-dd hh:mm:ss.ff3', 'yyyy-mm-dd hh:mm:ss.ff3+hh:mm', 'yyyy-mm-ddThh:mm:ss', 'yyyy-mm-ddThh:mm:ss+hh:mm', 'yyyy-mm-ddThh:mm:ss.ff3+hh:mm' ou 'yyyy-mm-ddThh:mm:ss.ff3+hh:mm'. Les fonctions macros qui renvoient une chaîne représentée sous la forme d'un horodatage avec fuseau horaire affiche une

précision de 9 chiffres pour la partie fractionnaire des secondes par défaut. Les options de format permettent un ajustement à une précision de 3 ou 0.

### Syntaxe

```
timestampMask ( string_expression1 , string_expression2 )
```

#### Exemple 1

```
# timestampMask ( $current_timestamp , 'yyyy-dd-mm' ) #
```

Résultat

2005-11-01

#### Exemple 2

```
# timestampMask ( '2005-11-01 12:00:00.000-05:00' , 'yyyy-mm-dd  
hh:mm:ss+hh:mm' ) #
```

Résultat

2005-11-01 12:00:00-05:00

#### Exemple 3

```
# timestampMask ( '2005-11-01 12:00:00.123456789-05:00' ,  
'yyyy-mm-ddThh:mm:ss+hh:mm.ff3+hh:mm' ) #
```

Résultat

2005-11-01T12:00:00.123-05:00

## toLocal

Renvoie la chaîne représentant un horodatage avec fuseau horaire résultant de l'ajustement de "expression\_chaîne" par rapport au fuseau horaire du système d'exploitation. Notez que la fonction macro timestampMask peut être utilisée pour ajuster cette donnée en sortie.

### Syntaxe

```
toLocal ( string_expression )
```

#### Exemple 1

```
# toLocal ( '2005-11-01 17:00:00.000-00:00' ) # where OS local time  
zone is -05:00
```

Résultat

2005-11-01 12:00:00.000000000-05:00

#### Exemple 2

```
# timestampMask ( toLocal ( '2005-11-01 17:00:00.000-00:00' ) ,  
'yyyy-mm-dd hh:mm:ss+hh:mm' ) # where OS local time zone is -05:00
```

Résultat

2005-11-01 12:00:00-05:00

### Exemple 3

```
# toLocal ( '2005-11-01 13:30:00.000-03:30' ) # where OS local time  
zone is -05:00
```

Résultat

```
2005-11-01 12:00:00.000000000-05:00
```

## tolower

Renvoie la chaîne "expression\_chaîne" dans laquelle tous les caractères sont convertis en minuscules selon les règles de l'environnement local "chaîne\_environnement\_local". Si aucun environnement local n'est spécifié, l'environnement local 'en' est utilisé.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
tolower ( string_expression [ , locale_string ] )
```

### Exemple 1

```
# tolower ( 'ABC' ) #
```

Résultat

```
abc
```

### Exemple 2

```
# tolower ( 'ABC' , 'fr' ) #
```

Résultat

```
abc
```

## toupper

Renvoie la chaîne "expression\_chaîne" dans laquelle tous les caractères sont convertis en majuscules selon les règles de l'environnement local "chaîne\_environnement\_local". Si le paramètre "chaîne\_environnement\_local" n'est pas spécifié, l'environnement local 'en' est utilisé.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
toupper ( string_expression [ , locale_string ] )
```

### Exemple 1

```
# toupper ( 'abc' ) #
```

Résultat

```
ABC
```

## Exemple 2

```
# toupper ( 'abc' , 'fr' ) #
```

Résultat

ABC

## toUTC

Renvoie la chaîne représentant un horodatage avec fuseau horaire résultant de l'ajustement de "expression\_chaine" par rapport au point de référence zéro du temps universel coordonné (UTC), également appelé heure GMT. Notez que la fonction macro timestampMask peut être utilisée pour ajuster cette donnée en sortie.

### Syntaxe

```
toUTC ( string_expression )
```

### Exemple 1

```
# toUTC ( '2005-11-01 12:00:00.000-05:00' ) #
```

Résultat

2005-11-01 17:00:00.000000000-00:00

### Exemple 2

```
# timestampMask( toUTC ( '2005-11-01 12:00:00.000-05:00' ) ,  
'yyyy-mm-dd hh:mm:ss.ff3+hh:mm' ) #
```

Résultat

2005-11-01 17:00:00.000-00:00

### Exemple 3

```
# toUTC ( $current_timestamp ) #
```

Résultat

2005-11-01 17:00:00.000000000-00:00

## uniques

Supprime les entrées en double dans le tableau. L'ordre des éléments est conservé.

### Syntaxe

```
unique ( array_expression )
```

### Exemple

```
# csv ( unique ( array ( 's3', 'a', 's3', 'x' ) ) ) #
```

Résultat

's3', 'a', 'x'

## urlencode

La fonction URL encode l'argument saisi. Cette fonction est utile lors de la définition de chaînes de connexions XML.

### Syntaxe

```
urlencode ( prompt ( 'userValue' ) )
```

### Exemple

```
urlencode ( prompt ( 'some_val' ) )
```

Résultat

```
%27testValue%27
```

## CSVIdentityName

Utilise les informations d'identité de l'utilisateur authentifié en cours pour rechercher des valeurs dans la table de mappage des paramètres. Chaque élément individuel de l'identité de l'utilisateur (nom de compte, noms de groupe, noms de rôle) est utilisé en tant que clé dans la table de mappage des paramètres. La liste unique de valeurs qui est extraite de la table de mappage des paramètres est renvoyée sous forme de chaîne, où chaque valeur est délimitée par des guillemets simples et où les valeurs multiples sont séparées par des virgules.

### Syntaxe

```
CSVIdentityName ( %parameter_map_name [ , separator_string ] )
```

### Exemple

```
# CSVIdentityName ( %security_clearance_level_map ) #
```

Résultat

```
'level_500', 'level_501', 'level_700'
```

## CSVIdentityNameList

Renvoie les éléments de l'identité de l'utilisateur (nom de compte, noms de groupe et noms de rôle) sous forme d'une liste de chaînes. La liste unique de valeurs est renvoyée sous forme de chaîne, où chaque valeur est délimitée par des guillemets simples et où les valeurs multiples sont séparées par des virgules.

### Syntaxe

```
CSVIdentityNameList ( [ separator_string ] )
```

### Exemple

```
# CSVIdentityNameList ( ) #
```

Résultat

```
'Everyone', 'Report Administrators', 'Query User'
```

## CAMPassport

Renvoie le passeport Cognos® Access Manager.

## Syntaxe

```
CAMPassport ( )
```

## Exemple

```
# CAMPassport ( ) #
```

Résultat

```
111:98812d62-4fd4-037b-4354-26414cf7ebef:3677162321
```

## CAMIDList

Renvoie les éléments de l'identificateur Cognos® Access Manager (CAMID) de l'utilisateur, tels que le nom de compte, les noms de groupe ou les noms de rôle, sous forme d'une liste de valeurs séparées par des virgules.

## Syntaxe

```
CAMIDList ( [ separator_string ] )
```

## Exemple

```
#CAMIDList ( ) #
```

Résultat

```
CAMID ( "::Everyone" ) , CAMID ( ":Authors" ) , CAMID ( ":Query  
Users" ) , CAMID ( ":Consumers" ) , CAMID ( ":Metrics Authors" )
```

## CAMIDListForType

Renvoie un tableau des identificateurs Cognos® Access Manager (CAMID) de l'utilisateur selon le type d'identité (compte, groupe ou rôle). CAMIDListForType peut être utilisé avec des fonctions macros dans des fichiers CSV ou de jointure.

## Syntaxe

```
CAMIDListForType ( identity type )
```

## Exemple

```
[qs].[userRole] in ( # csv ( CAMIDListForType ( 'role' ) ) # )
```

Résultat

```
[qs].[userRole] in ( 'CAMID ( "::System Administrators" ) ' , 'CAMID  
( ":Authors" ) ' )
```

## CASE (simple)

Cette syntaxe de macro correspond au modèle d'une instruction CASE simple comprenant les fonctions case, when, then, else et end. Notez que cette syntaxe de macro est uniquement prise en charge en mode DQM.

## Syntaxe

```
CASE <expression> WHEN <literal> THEN <expression> [ELSE  
<expression>] END
```

## Exemple

```
#CASE prompt('pDateRange','token') WHEN 'Current Week' THEN '[PR  
Current Week]' ELSE '[PR Prior Week]' END#
```

Résultat

[PR Current Week]

---

## Fonctions communes

### abs

Renvoie la valeur absolue de "numeric\_expression". Les valeurs négatives sont renvoyées sous forme de valeurs positives.

### Syntaxe

```
abs ( numeric_expression )
```

### Exemple 1

```
abs ( 15 )
```

Résultat

15

### Exemple 2

```
abs ( -15 )
```

Résultat

15

### cast

Convertit "expression" en un type de données défini. Certains types de données permettent de définir une longueur et un degré de précision. Assurez-vous que les attributs de type et de taille de la cible sont corrects. Les types de données suivants peuvent être utilisés pour "datatype\_specification" : CHARACTER, VARCHAR, CHAR, NUMERIC, DECIMAL, INTEGER, BIGINT, SMALLINT, REAL, FLOAT, DATE, TIME, TIMESTAMP, TIME WITH TIME ZONE, TIMESTAMP WITH TIME ZONE et INTERVAL. Lorsque vous définissez une conversion de type INTERVAL, vous devez spécifier l'un des qualificatifs suivants : YEAR, MONTH, ou YEAR TO MONTH pour un intervalle de type Année à mois ; DAY, HOUR, MINUTE, SECOND, DAY TO HOUR, DAY TO MINUTE, DAY TO SECOND, HOUR TO MINUTE, HOUR TO SECOND ou MINUTE TO SECOND pour un intervalle de type Jour à seconde. Remarques : Lorsque vous convertissez une valeur de type TIMESTAMP à DATE, la partie heure de la valeur d'horodatage est omise. Lorsque vous convertissez une valeur de type TIMESTAMP à TIME, la partie date de la valeur d'horodatage est omise. Lorsque vous convertissez une valeur de type DATE à TIMESTAMP, la partie heure de la valeur d'horodatage est définie sur zéro. Lorsque vous convertissez une valeur de type TIME à TIMESTAMP, la partie date de la valeur d'horodatage est définie à la date en cours du système. Il est impossible de convertir un type d'intervalle en un autre (parce que le nombre de jours dans un mois est variable, par exemple). Notez que vous pouvez seulement définir le nombre de chiffres pour le premier qualificatif, par exemple, YEAR(4) TO

MONTH ou DAY(5). Des erreurs seront générées si les attributs de taille et de type de la cible ne sont pas compatibles avec ceux de la source.

### **Syntaxe**

```
cast ( expression , datatype_specification )
```

### **Exemple 1**

```
cast ( '123' , integer )
```

Résultat

123

### **Exemple 2**

```
cast ( 12345 , varchar ( 10 ) )
```

Résultat

Une chaîne contenant 12345

## **ceil**

Renvoie le plus petit entier supérieur ou égal à "numeric\_expression".

### **Syntaxe**

```
ceil ( numeric_expression )
```

## **ceiling**

Renvoie le plus petit entier supérieur ou égal à "numeric\_expression".

### **Syntaxe**

```
ceiling ( numeric_expression )
```

### **Exemple 1**

```
ceiling ( 4.22 )
```

Résultat

5

### **Exemple 2**

```
ceiling ( -1.23 )
```

Résultat

-1

## **char\_length**

Renvoie le nombre de caractères logiques contenus dans "string\_expression". Le nombre de caractères logiques peut être différent du nombre d'octets dans certaines langues d'Asie orientale.

### **Syntaxe**

```
char_length ( string_expression )
```



### **Exemple**

```
char_length ( 'Canada' )
```

Résultat

6

## **character\_length**

Renvoie le nombre de caractères contenus dans "string\_expression".

### **Syntaxe**

```
character_length ( string_expression )
```

### **Exemple**

```
character_length ( 'Canada' )
```

Résultat

6

## **coalesce**

Renvoie le premier argument non nul (ou la valeur NULL si tous les arguments sont nuls). Nécessite deux arguments ou davantage dans "expression\_list".

### **Syntaxe**

```
coalesce ( expression_list )
```

### **Exemple**

```
coalesce ( [Unit price], [Unit sale price] )
```

Résultat

Renvoie le prix unitaire, ou le prix unitaire de vente si le premier est nul.

## **current\_date**

Renvoie une valeur de date représentant la date en cours de l'ordinateur sur lequel tourne le logiciel de base de données.

### **Syntaxe**

```
current_date
```

### **Exemple**

```
current_date
```

Résultat

2003-03-04

## **current\_time**

Renvoie une valeur d'heure avec fuseau horaire représentant l'heure courante de l'ordinateur sur lequel s'exécute le logiciel de base de données, si cette fonction est prise en charge dans la base de données. Sinon, représente l'heure de l'ordinateur exécutant le logiciel IBM Cognos BI.

## Syntaxe

`current_time`

## Exemple

`current_time`

Résultat

16:33:11.354+05:00

## current\_timestamp

Renvoie une valeur de date-heure avec fuseau horaire représentant l'heure courante de l'ordinateur sur lequel s'exécute le logiciel de base de données, si cette fonction est prise en charge dans la base de données. Sinon, représente l'heure de l'ordinateur exécutant le logiciel IBM Cognos BI.

## Syntaxe

`current_timestamp`

## Exemple

`current_timestamp`

Résultat

2003-03-03 16:40:15.535+05:00

## exp

Renvoie "e" à la puissance "numeric\_expression". La constante "e" est la base du logarithme naturel.

## Syntaxe

`exp ( numeric_expression )`

## Exemple

`exp ( 2 )`

Résultat

7,389056

## extract

Renvoie un entier représentant la valeur de "datepart" (année, mois, jour, heure, minute, seconde) dans "datetime\_expression".

## Syntaxe

`extract ( datepart , datetime_expression )`

## Exemple 1

`extract ( year , 2003-03-03 16:40:15.535 )`

Résultat

2003

### Exemple 2

```
extract ( hour , 2003-03-03 16:40:15.535 )
```

Résultat

16

## floor

Renvoie le plus grand entier inférieur ou égal à "numeric\_expression".

### Syntaxe

```
floor ( numeric_expression )
```

### Exemple 1

```
floor ( 3.22 )
```

Résultat

3

### Exemple 2

```
floor ( -1.23 )
```

Résultat

-2

## ln

Renvoie le logarithme naturel de "numeric\_expression".

### Syntaxe

```
ln ( numeric_expression )
```

### Exemple

```
ln ( 4 )
```

Résultat

1,38629

## localtime

Renvoie une valeur temporelle représentant l'heure en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

### Syntaxe

```
localtime
```

### Exemple

```
localtime
```

Résultat

16:33:11

## localtimestamp

Renvoie une valeur de date-heure représentant l'horodatage en cours de l'ordinateur sur lequel s'exécute le logiciel de base de données.

### Syntaxe

```
localtimestamp
```

### Exemple

```
localtimestamp
```

Résultat

```
2003-03-03 16:40:15
```

## lower

Convertit en minuscules tous les caractères en majuscules de "string\_expression".

### Syntaxe

```
lower ( string_expression )
```

### Exemple

```
lower ( 'ABCDEF' )
```

Résultat

```
abcdef
```

## mod

Renvoie le reste (modulo) de "integer\_expression1" divisé par "integer\_expression2". La valeur "integer\_expression2" doit être différente de zéro, sinon cela génère une condition d'exception.

### Syntaxe

```
mod ( integer_expression1, integer_expression2 )
```

### Exemple

```
mod ( 20 , 3 )
```

Résultat

```
2
```

## nullif

Renvoie la valeur Null si les expressions "expression1" et "expression2" sont égales. Sinon la fonction renvoie "expression1".

### Syntaxe

```
nullif ( expression1, expression2 )
```

## octet\_length

Renvoie le nombre d'octets contenus dans "string\_expression".

## Syntaxe

```
octet_length ( string_expression )
```

### Exemple 1

```
octet_length ( 'ABCDEF' )
```

Résultat

6

### Exemple 2

```
octet_length ( '' )
```

Résultat

0

## position

Renvoie la valeur de l'entier représentant la position de début de "string\_expression1" dans "string\_expression2" ou 0 lorsque "string\_expression1" est introuvable.

## Syntaxe

```
position ( string_expression1 , string_expression2 )
```

### Exemple 1

```
position ( 'C' , 'ABCDEF' )
```

Résultat

3

### Exemple 2

```
position ( 'H' , 'ABCDEF' )
```

Résultat

0

## power

Renvoie "numeric\_expression1" élevé à la puissance "numeric\_expression2". Si l'expression "numeric\_expression1" est négative, "numeric\_expression2" doit représenter un entier.

## Syntaxe

```
power ( numeric_expression1 , numeric_expression2 )
```

### Exemple

```
power ( 3 , 2 )
```

Résultat

9

## **\_round**

Renvoie l'expression "expression\_numérique" arrondie à "expression\_entier" positions à droite du séparateur décimal. Remarques : "expression\_entier" doit être un entier non négatif. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### **Syntaxe**

```
_round ( numeric_expression , integer_expression )
```

### **Exemple**

```
_round ( 1220.42369, 2 )
```

Résultat

1220.42

## **row**

Le constructeur row représente une collection de valeurs organisée comme une ligne de données. Il est utilisable dans des expressions conditionnelles (par exemple : IF-THEN-ELSE) et dans des expressions de filtre (par exemple : la clause IN).

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
row ( expression_list )
```

### **Exemple 1**

```
if ( row([RetailerName],[OrderMethodCode]) = row('ActiForme',4) )
```

```
  then ('A')
  else ('B')
```

Résultat

Renvoie "A" si le nom du détaillant est "ActiForme" et le code du mode de commande est 4. Sinon, la valeur "B" est renvoyée.

### **Exemple 2**

```
case row([RetailerName],[OrderMethodCode])
  when row('Advanced
Climbing Ltd',3) then 1
  when row('ActiForme',5) then 2
  else 3
end
```

Résultat

Renvoie 1 si le nom du détaillant est "Advanced Climbing Ltd" et le code du mode de commande est 3. Renvoie 2 si le nom du détaillant est "ActiForme" et le code du mode de commande est 5. Sinon, la valeur "3" est renvoyée.

### Exemple 3

```
row ( [OrderMethodCode], [Year] ) in ( [Query].[OMC], [Query].[YR] )
```

Résultat

Les données renvoyées sont filtrées à l'aide des deux conditions suivantes : 1) [OrderMethodCode] dans ([Query].[OMC]) 2) [Year] dans ([Query].[YR])

## sqrt

Renvoie la racine carrée de "numeric\_expression". La valeur "expression\_numérique" ne doit pas être négative.

### Syntaxe

```
sqrt ( numeric_expression )
```

### Exemple

```
sqrt ( 9 )
```

Résultat

3

## substring

Renvoie la sous-chaîne de "string\_expression" qui commence à la position "integer\_expression1" et dont la longueur est de "integer\_expression2" caractères, ou qui se termine par "string\_expression" si l'expression "integer\_expression2" est omise. Le premier caractère de "string\_expression" occupe la position 1.

### Syntaxe

```
substring ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

### Exemple

```
substring ( 'abcdefg' , 3 , 2 )
```

Résultat

cd

## trim

Renvoie une expression "string\_expression" sans les blancs de droite ou de gauche ou sans le caractère indiqué dans "match\_character\_expression". "both" est la valeur implicite lorsque le premier argument n'est pas indiqué et un blanc est la valeur implicite lorsque le second argument n'est pas indiqué.

### Syntaxe

```
trim ( [ [ trailing|leading|both ] [ match_character_expression ] , ]  
string_expression )
```

### Exemple 1

```
trim ( trailing 'A' , 'ABCDEFA' )
```

Résultat

ABCDEF

### Exemple 2

```
trim ( both , ' ABCDEF ' )
```

Résultat

ABCDEF

## upper

Convertit tous les caractères minuscules de "string\_expression" en majuscules.

### Syntaxe

```
upper ( string_expression )
```

### Exemple

```
upper ( 'abcdef' )
```

Résultat

ABCDEF

## width-bucket

Pour une expression donnée, cette fonction renvoie le numéro du compartiment dans lequel la valeur de l'expression tomberait après son évaluation.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### Syntaxe

```
width-bucket ( numeric_expression , min_value , max_value ,  
num_of_buckets )
```

### Exemple

```
width-bucket ( Quantity , 100 , 5000 , 10 )
```

Résultat

Pour chaque ligne, renvoie le numéro du compartiment (de 0 à 11) pour la quantité en cours.

Données du résultat

Quantity	width-bucket (Quantity)
50	0
450	1
1400	3
3600	8
4900	10
5000	11



## Fonctions trigonométriques

### **arccos**

Cette fonction trigonométrique inverse renvoie l'arc cosinus de l'argument, où l'argument est une valeur entre -1 et 1 et le résultat est une valeur exprimée en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
arccos ( numeric_expression )
```

### **Exemple**

```
arccos ( -1 )
```

Résultat

3.1415

### **arcsin**

Cette fonction trigonométrique inverse renvoie l'arc sinus de l'argument, où l'argument est une valeur entre -1 et 1 et le résultat est une valeur exprimée en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
arcsin ( numeric_expression )
```

### **Exemple**

```
arcsin ( 0 )
```

Résultat

3.1415

### **arctan**

Cette fonction trigonométrique inverse renvoie l'arc tangente de l'argument, où l'argument est une valeur entre -1 et 1 et le résultat est une valeur exprimée en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

### **Syntaxe**

```
arctan ( numeric_expression )
```

### **Exemple**

```
arctan ( 0 )
```

Résultat

3.1415

**cos**

Cette fonction trigonométrique renvoie le cosinus de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

```
cos ( numeric_expression )
```

**Exemple**

```
cos ( 0.3333 * 3.1415 )
```

Résultat

0.5

**coshyp**

Cette fonction trigonométrique renvoie le cosinus hyperbolique de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

```
coshyp ( numeric_expression )
```

**Exemple**

```
coshyp ( 0 )
```

Résultat

1

**sin**

Cette fonction trigonométrique renvoie le sinus de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

```
sin ( numeric_expression )
```

**Exemple**

```
sin ( 0.1667 * 3.1415 )
```

Résultat

0.5

**sinhyp**

Cette fonction trigonométrique renvoie le sinus hyperbolique de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

`sinhyp ( numeric_expression )`

**Exemple**

`sinhyp ( 0 )`

Résultat

0

**tan**

Cette fonction trigonométrique renvoie la tangente de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

`tan ( numeric_expression )`

**Exemple**

`tan ( 0.25 * 3.1415 )`

Résultat

1

**tanhyp**

Cette fonction trigonométrique renvoie la tangente hyperbolique de l'argument, où l'argument est un angle exprimé en radians.

Remarque : il est uniquement pris en charge dans le mode de requête dynamique.

**Syntaxe**

`tanhyp ( numeric_expression )`

**Exemple**

`tanhyp ( 0 )`

Résultat

0

---

## Fonctions dimensionnelles

**ancestor**

Renvoie l'ancêtre de "membre" au niveau "niveau" ou à "entier" niveaux au-dessus de "membre". Remarque : nous ne pouvons pas garantir que les résultats seront cohérents s'il y a plusieurs ancêtres de cette nature.

**Syntaxe**

`ancestor ( member, level|integer )`

### Exemple 1

```
ancestor ( [TrailChef Water Bag] , 1 )
```

Résultat

Ustensiles de cuisine

### Exemple 2

```
ancestor ( [TrailChef Water Bag] , 2 )
```

Résultat

Matériel de camping

### Exemple 3

```
ancestor ( [TrailChef Water Bag] , [great_outdoors_company].  
[Products].[Products].[Product type] )
```

Résultat

Ustensiles de cuisine

## ancestors

Renvoie tous les ancêtres de "membre" à un niveau "niveau" ou à une distance "index" au-dessus du membre. (La plupart des sources de données ne prennent en charge qu'un ancêtre à un niveau donné. Si la source de données prend en charge plusieurs ancêtres, le résultat est un ensemble de données).

### Syntaxe

```
ancestors ( member , level|index )
```

### Exemple 1

```
ancestors ( [TrailChef Water Bag] , 1 )
```

Résultat

Ustensiles de cuisine

### Exemple 2

```
ancestors ( [TrailChef Water Bag] , 2 )
```

Résultat

Matériel de camping

### Exemple 3

```
ancestors ( [TrailChef Water Bag] , [great_outdoors_company].  
[Products].[Products].[Product type] )
```

Résultat

Ustensiles de cuisine

## bottomCount

Trie un ensemble en fonction de la valeur de "expression\_numérique" évaluée pour chacun des membres de "expression\_ensemble" et renvoie les "expression\_index" derniers membres.

### Syntaxe

```
bottomCount ( set_expression , index_expression , numeric_expression )
```

### Exemple

```
bottomCount ( [great_outdoors_company].[Products].[Products].[Product  
line] , 2 , [Revenue] )
```

Résultat

Renvoie les deux derniers membres de l'ensemble trié en fonction du revenu.

Données du résultat

Prod line	Revenue
Camping	\$3,171,114.92
Personal	\$20,891,350.60

## bottomPercent

Trie par ordre croissant l'ensemble indiqué dans "expression\_ensemble" et renvoie les éléments du niveau le plus bas à partir de l'ensemble trié dont le pourcentage cumulatif du total est supérieur ou égal à "pourcentage".

### Syntaxe

```
bottomPercent ( set_expression , percentage , numeric_expression )
```

### Exemple

```
bottomPercent ( set ( [Camping Equipment] , [Golf Equipment] ,  
[Mountaineering Equipment] ) , 40 , [2006] )
```

Résultat

Pour l'ensemble Matériel de camping, Matériel de golf et Matériel de montagne, renvoie les membres ayant le bénéfice brut le moins élevé atteignant un total au moins égal à 40 % du total général pour l'année 2006.

## bottomSum

Trie par ordre croissant l'ensemble indiqué dans "expression\_ensemble" et renvoie les éléments du niveau le plus bas à partir de l'ensemble trié dont le total cumulatif est supérieur ou égal à "valeur".

### Syntaxe

```
bottomSum ( set_expression , value , numeric_expression )
```

### Exemple

```
bottomSum ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) , 6000000 , tuple ( [2006] ,  
[great_outdoors_company].[Measures].[Gross profit] ) )
```

Résultat

Pour les membres de la ligne de produits, renvoie les membres ayant le bénéfice brut le moins élevé atteignant un total au moins égal à 6 000 000 \$ pour l'année 2006.

## caption

Renvoie les valeurs de légende de "niveau", "membre" ou "expression\_ensemble". La légende constitue le nom d'affichage de chaîne d'un élément ; elle ne correspond pas forcément à l'identificateur unique utilisé pour générer la clé métier ou le nom unique de membre de l'élément. La légende n'est pas nécessairement unique ; par exemple, la légende d'un mois peut renvoyer le nom du mois sans aucun détail relatif à l'année visant à rendre la valeur unique.

### Syntaxe

```
caption ( level|member|set_expression )
```

### Exemple 1

```
caption ( [TrailChef Water Bag] )
```

Résultat

TrailChef Water Bag

### Exemple 2

```
caption ( [great_outdoors_company].[Products].[Products].[Product  
line] )
```

Résultat

Renvoie les valeurs de légende de l'ensemble Ligne de produits.

Données du résultat

Camping Equipment  
Mountaineering Equipment  
Personal Accessories  
Outdoor Protection  
Golf Equipment

## children

Renvoie l'ensemble d'enfants d'un membre spécifié.

### Syntaxe

```
children ( member )
```

### Exemple

```
children ( [Camping Equipment] )
```

Résultat

Renvoie l'ensemble d'enfants de l'ensemble Matériel de camping.

Données du résultat

Cooking Gear  
Tents  
Sleeping Bags  
Packs  
Lanterns

## closingPeriod

Renvoie le dernier membre de même niveau parmi les descendants d'un membre situé au niveau "niveau". Cette fonction est généralement utilisée avec une dimension de temps.

### Syntaxe

```
closingPeriod ( level [ , member ] )
```

### Exemple 1

```
closingPeriod ( [great_outdoors_company].[Years].[Years].[Month] )
```

Résultat

2006/Déc

### Exemple 2

```
closingPeriod ( [great_outdoors_company].[Years].[Years].[Year] )
```

Résultat

2006

### Exemple 3

```
closingPeriod ( [great_outdoors_company].[Years].[Years].[Month] ,  
[2006 Q 4] )
```

Résultat

2006/Déc

## cousin

Renvoie le membre enfant de "membre2" avec la même position relative que "membre1" par rapport à son parent. Cette fonction apparaît dans l'exemple de rapport Revenus par filiale VA, 2005, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
cousin ( member1 , member2 )
```

### Exemple 1

```
cousin ( [Irons] , [Camping Equipment] )
```

Résultat

Ustensiles de cuisine

## Exemple 2

cousin ( [Putters] , [Camping Equipment] )

Résultat

Sacs de couchage

## completeTuple

Indique l'emplacement d'une cellule (intersection) en fonction des membres spécifiés provenant tous de dimensions différentes. Toutefois, completeTuple () inclut implicitement le membre par défaut de toutes les dimensions non définies ailleurs dans les arguments, plutôt que le membre en cours. Si aucune mesure n'est définie dans la fonction CompleteTuple, celle-ci utilise la mesure par défaut et non la valeur currentMeasure de la requête. Cette fonction apparaît dans l'exemple de rapport Effectifs prévus, accessible dans le module Entrepôt de données VA (analyse).

## Syntaxe

completeTuple ( member { , member } )

## Exemple 1

completeTuple ( [Mountaineering Equipment] , [Fax] )

Résultat

Par défaut, la fonction completeTuple n'utilise pas la valeur currentMember, contrairement à la fonction tuple. Les valeurs contenues dans la première colonne sont identiques d'année en année, car le membre par défaut de la dimension Années, le membre racine, est utilisé à la place du membre en cours. De même, la première colonne affiche les revenus plutôt que la quantité vendue, car la mesure Revenus est définie par défaut dans la dimension Mesures. Si aucune mesure n'est définie dans la fonction CompleteTuple, celle-ci utilise la mesure par défaut et non la valeur currentMeasure de la requête.

Données du résultat

Qty Sold	Mountaineering Sales by Fax
2004	\$1,220,329.38
2005	\$1,220,329.38
2006	\$1,220,329.38

## Exemple 2

completeTuple ( [Mountaineering Equipment] , [Fax] , [Quantity sold] ,  
currentMember ( [great\_outdoors\_company].[Years].[Years] ) )

Résultat

La fonction completeTuple utilise la valeur currentMember de la dimension Années et la mesure Quantité vendue.

Données du résultat

Qty Sold	Mountaineering Sales by Fax
2004	0



**Qty Sold**  
2005  
2006

**Mountaineering Sales by Fax**  
8,746  
7,860

## currentMember

Renvoie le membre en cours de la hiérarchie pendant une itération. Si "hiérarchie" n'est pas présent dans le contexte dans lequel l'expression est évaluée, son membre par défaut est utilisé. Cette fonction apparaît dans l'exemple de rapport interactif Moyennes continues et mobiles.

### Syntaxe

`currentMember ( hierarchy )`

## defaultMember

Renvoie le membre par défaut de "hiérarchie".

### Syntaxe

`defaultMember ( hierarchy )`

### Exemple 1

`defaultMember ( [great_outdoors_company].[Products].[Products] )`

Résultat

Produits

### Exemple 2

`defaultMember ( [great_outdoors_company].[Years].[Years] )`

Résultat

Année

### Exemple 3

`defaultMember ( hierarchy ( [great_outdoors_company].[Measures].[Quantity sold] ) )`

Résultat

Revenus

## descendants

Renvoie l'ensemble de descendants de "membre" ou "expression\_ensemble" situé au niveau "niveau" (nom qualifié) ou à la distance "distance" (nombre entier compris entre 0 et n) de la racine. Plusieurs options peuvent être spécifiées (séparées par une espace) afin de déterminer les membres qui sont renvoyés. `self` : seuls les membres qui figurent au niveau spécifié sont inclus dans l'ensemble final (comportement par défaut en l'absence d'options). `before` : s'il existe des niveaux intermédiaires entre le niveau du membre et celui qui est spécifié, les membres de ces niveaux sont inclus. Si le niveau indiqué est le même que celui du membre auquel la fonction est appliquée, le membre est inclus dans l'ensemble final. `beforewithmember` : s'il existe des niveaux intermédiaires entre le niveau du membre et celui qui est spécifié, les membres de ces niveaux sont inclus. Le

membre auquel la fonction est appliquée est également inclus dans l'ensemble final. after : s'il existe d'autres niveaux après le niveau indiqué, les membres de ces niveaux sont inclus dans l'ensemble final. Cette fonction apparaît dans l'exemple de rapport Commissions sur les ventes pour l'Europe centrale, accessible dans le module Entrepôt de données VA (analyse).

## Syntaxe

```
descendants ( member|set_expression , level|distance [ ,  
  { self|before|beforewithmember|after } ] )
```

## Exemple 1

```
descendants ( [great_outdoors_company].[Products].[Products].  
[Products] , [great_outdoors_company].[Products].[Products].[Product  
type] )
```

### Résultat

Renvoie l'ensemble des descendants de l'ensemble Products au niveau Type de produit. Remarque : [great\_outdoors\_company].[Products].[Products].[Products] constitue le membre racine de la hiérarchie Products.

### Données du résultat

```
Cooking Gear  
Sleeping Bags  
Packs  
Tents  
...  
Eyewear  
Knives  
Watches
```

## Exemple 2

```
descendants ( [great_outdoors_company].[Products].[Products].  
[Products] , 1 )
```

### Résultat

Renvoie l'ensemble des descendants de l'ensemble Products au premier niveau.

### Données du résultat

```
Camping Equipment  
Golf Equipment  
Mountaineering Equipment  
Outdoor Protection  
Personal Accessories
```

## Exemple 3

```
descendants ( [great_outdoors_company].[Products].[Products].  
[Products] , 3 , before )
```

### Résultat

Renvoie les descendants de l'ensemble Products avant le troisième niveau.

### Données du résultat

```
Camping Equipment  
Cooking Gear  
Sleeping Bags
```

Packs  
Tents  
...  
Eyewear  
Knives  
Watches

#### Exemple 4

```
descendants ( [great_outdoors_company].[Products].[Products].  
[Products] , 2 , self before )
```

Résultat

Renvoie l'ensemble des descendants de l'ensemble Products avant le deuxième niveau (inclus).

Données du résultat

Camping Equipment  
Cooking Gear  
Sleeping Bags  
Packs  
Tents  
...  
Eyewear  
Knives  
Watches

## except

Renvoie les membres de "expression\_ensemble1" qui ne sont pas inclus aussi dans "expression\_ensemble2". Les doubles sont conservés lorsque le mot clé facultatif "all" est utilisé comme troisième argument.

### Syntaxe

```
except ( set_expression1 , set_expression2 [ , all ] )
```

### Exemple

```
except ( set ( [Camping Equipment] , [Mountaineering Equipment] ) ,  
set ( [Camping Equipment] , [Golf Equipment] ) )
```

Résultat

Matériel de montagne

## filter

Retourne un ensemble résultant du filtrage d'un ensemble indiqué en fonction d'une condition booléenne. Chaque membre est inclus dans le résultat si et seulement si la valeur correspondante de 'expression\_booléenne' est True.

### Syntaxe

```
filter ( set_expression , Boolean_expression )
```

### Exemple

```
filter ( [Product line] , [Gross margin] > .30 )
```

Résultat

Matériel de montagne

## firstChild

Renvoie le premier enfant de "membre".

### Syntaxe

```
firstChild ( member )
```

### Exemple 1

```
firstChild ( [By Product Lines] )
```

Résultat

Matériel de camping

### Exemple 2

```
firstChild ( [Camping Equipment] )
```

Résultat

Ustensiles de cuisine

## firstSibling

Renvoie le premier enfant du parent de "membre".

### Syntaxe

```
firstSibling ( member )
```

### Exemple 1

```
firstSibling ( [Outdoor Protection] )
```

Résultat

Matériel de camping

### Exemple 2

```
firstSibling ( [Camping Equipment] )
```

Résultat

Matériel de camping

## \_format

Associe un format à l'expression. mot clé `_format` peut avoir la valeur `PERCENTAGE_0`, `PERCENTAGE_1` ou `PERCENTAGE_2`. `PERCENTAGE_1` renvoie un pourcentage comportant un seul chiffre après la virgule, `PERCENTAGE_2` renvoie un pourcentage comportant deux chiffres après la virgule et `PERCENTAGE_3` renvoie une valeur de pourcentage comportant jusqu'à trois chiffres après la virgule (par exemple, 0,965).

### Syntaxe

```
_format ( expression , format_keyword )
```

## Exemple

```
_format ( [Unit Sale Price] / [Unit Price] , PERCENTAGE_2 )
```

Résultat

75,12%

## emptySet

Renvoie un membre vide pour "hiérarchie". Cet élément est généralement utilisé en tant que paramètre fictif lors du développement ou de la conception de rapports dynamiques (avec IBM® Cognos® Software Development Kit ou via la conception de rapports). En créant une donnée élémentaire qui contient la fonction emptyset, vous pouvez créer des expressions complexes que vous pouvez modifier par la suite en redéfinissant la donnée élémentaire emptyset.

## Syntaxe

```
emptySet ( hierarchy )
```

## Exemple

```
except ( [great_outdoors_company].[Products].[Products].[Product  
line] , emptySet ( [great_outdoors_company].[Products].[Products] ) )
```

Résultat

Renvoie l'ensemble Ligne de produits et un ensemble vide pour l'ensemble Produits.

Données du résultat

```
Camping Equipment  
Golf Equipment  
Mountaineering Equipment  
Outdoor Protection  
Personal Accessories
```

## generate

Evalue "expression\_ensemble2" pour chaque membre de "expression\_ensemble1" et effectue l'union des ensembles résultants. Le résultat conserve les doublons uniquement lorsque le mot clé facultatif "all" est fourni comme troisième argument.

## Syntaxe

```
generate ( set_expression1 , set_expression2 [ , all ] )
```

## Exemple

```
generate ( [Product line] , topCount ( descendants ( currentMember ( [great_outdoors_company].[Products].[Products] ) , [great_outdoors_company].[Products].[Products].[Product name] ) , 2 , [Revenue] ) )
```

Résultat

Renvoie les deux produits les plus vendus pour chaque ligne de produits, classés par revenus.

## head

Renvoie les "expression\_index" premiers éléments de "expression\_ensemble". La valeur par défaut de "expression\_index" est 1.

### Syntaxe

```
head ( set_expression [ , index_expression ] )
```

### Exemple 1

```
head ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) )
```

Résultat

Matériel de camping

### Exemple 2

```
head ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) , 2 )
```

Résultat

Renvoie les deux premiers membres de l'ensemble Ligne de produits.

Données du résultat

Camping Equipment  
Mountaineering Equipment

## hierarchize

Classe les membres de "expression\_ensemble" dans une hiérarchie. Les membres d'un niveau sont classés dans l'ordre naturel. Il s'agit du classement par défaut des membres d'une dimension si aucun autre critère de tri n'est indiqué.

### Syntaxe

```
hierarchize ( set_expression )
```

### Exemple

```
hierarchize ( set ( [Golf Equipment] , [Mountaineering Equipment] ,  
[Camping Equipment] ) )
```

Résultat

Renvoie Matériel de camping, Matériel de golf, Matériel de montagne.

## hierarchy

Renvoie la hiérarchie contenant "niveau", "membre" ou "expression\_ensemble".

### Syntaxe

```
hierarchy ( level|member|set_expression )
```

### Exemple 1

```
hierarchy ( [Cooking Gear] )
```

Résultat

Renvoie tous les membres de la hiérarchie contenant Ustensiles de cuisine.

Données du résultat

```
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
...
Mountain Man Extreme
Mountain Man Deluxe
```

## Exemple 2

```
hierarchy ( [great_outdoors_company].[Products].[Products].[Product
line] )
```

Résultat

Renvoie tous les membres de la hiérarchie contenant Ligne de produits.

Données du résultat

```
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
...
Mountain Man Extreme
Mountain Man Deluxe
```

## item

Renvoie un membre à partir d'un emplacement "index" dans "expression\_ensemble". L'index de l'ensemble commence à zéro.

### Syntaxe

```
item ( set_expression , index )
```

### Exemple

```
item ( children ( [Camping Equipment] ) , 2 )
```

Résultat

Sacs de couchage

## intersect

Renvoie l'intersection de "expression\_ensemble1" et "expression\_ensemble2". Le résultat conserve les doublons uniquement lorsque le mot clé facultatif "all" est fourni comme troisième argument.

### Syntaxe

```
intersect ( set_expression1 , set_expression2 [ , all ] )
```

### Exemple

```
intersect ( set ( [Camping Equipment] , [Mountaineering Equipment] ) ,
set ( [Camping Equipment] , [Outdoor Protection] , ) , all )
```

Résultat

Matériel de camping

## lag

Renvoie le membre de même niveau qui se trouve à "expression\_index" positions avant "membre".

### Syntaxe

```
lag ( member , index_expression )
```

### Exemple 1

```
lag ( [Tents] , 1 )
```

Résultat

Ustensiles de cuisine

### Exemple 2

```
lag ( [Tents] , -2 )
```

Résultat

Sacs à dos et autres contenants

## lastChild

Renvoie le dernier enfant d'un membre spécifié.

### Syntaxe

```
lastChild ( member )
```

### Exemple 1

```
lastChild ( Cooking Gear )
```

Résultat

Ustensiles ChefDeCamp

### Exemple 2

```
lastChild ( [By Product Line] )
```

Résultat

Matériel de golf

## lastPeriods

Renvoie l'ensemble de membres d'un même niveau qui finit par "membre". Le nombre de membres renvoyés correspond à la valeur absolue de "expression\_entier". Si "expression\_entier" est une valeur négative, les membres situés après le membre spécifié, y compris celui-ci, sont renvoyés. Cette fonction est généralement utilisée avec une dimension de temps. Cette fonction apparaît dans l'exemple de rapport interactif Moyennes continues et mobiles.



## Syntaxe

`lastPeriods ( integer_expression , member )`

### Exemple 1

`lastPeriods ( 2 , [2006 Q 4] )`

Résultat

Renvoie les deux derniers membres du niveau se terminant par 2006 Q 4.

Données du résultat

2006 Q 3  
2006 Q 4

### Exemple 2

`lastPeriods ( -3 , [2006 Q 4] )`

Résultat

Renvoie les trois derniers membres du niveau débutant par 2006 Q 4.

Données du résultat

2006 Q 4  
2007 Q 1  
2007 Q 2

## lastSibling

Renvoie le dernier enfant du parent d'un membre spécifié.

### Syntaxe

`lastSibling ( member )`

### Exemple

`lastSibling ( [Camping Equipment] )`

Résultat

Matériel de golf

## lead

Renvoie le membre de même niveau qui se trouve à "expression\_index" positions après "membre". Si "expression\_index" est une valeur négative, renvoie le membre de même niveau qui se trouve à "expression\_index" positions avant "membre".

### Syntaxe

`lead ( member , index_expression )`

### Exemple 1

`lead ( [Outdoor Protection] , 1 )`

Résultat

Accessoires personnels

## Exemple 2

```
lead ( [Outdoor Protection] , -2 )
```

Résultat

Matériel de golf

## level

Renvoie le niveau de "membre".

### Syntaxe

```
level ( member )
```

### Exemple

```
level ( [Golf Equipment] )
```

Résultat

Renvoie les membres du niveau Matériel de golf.

Données du résultat

```
Camping Equipment  
Mountaineering Equipment  
Personal Accessories  
Outdoor Protection  
Golf Equipment
```

## levels

Retourne le niveau de "hiérarchie" dont la distance à la racine est indiquée par "index".

### Syntaxe

```
levels ( hierarchy , index )
```

### Exemple 1

```
levels ( [great_outdoors_company].[Products].[Products] , 2 )
```

Résultat

Renvoie les membres situés à deux niveaux de la hiérarchie racine Produits.

Données du résultat

```
Cooking Gear  
Sleeping Bags  
Packs  
Tents  
...  
Irons  
Putters  
Woods  
Golf Accessories
```

### Exemple 2

```
levels ( [great_outdoors_company].[Products].[Products] , 1 )
```

Résultat

Renvoie les membres situés à un niveau de la hiérarchie racine Produits.

Données du résultat

Camping Equipment  
Mountaineering Equipment  
Personal Accessories  
Outdoor Protection  
Golf Equipment

## linkMember

Retourne le membre correspondant dans le "niveau" ou la "hiérarchie" (de la même dimension). Pour les hiérarchies avec niveaux, un niveau doit être indiqué comme second argument, et pour les hiérarchies parent-enfant, une hiérarchie doit être spécifiée. Une exception se produit lorsque le second paramètre ne produit pas une hiérarchie de la dimension du membre. Notez que les membres calculés ne sont pas pris en charge comme premier argument.

### Syntaxe

```
linkMember ( member , level|hierarchy )
```

## members

Renvoie l'ensemble des membres contenus dans l'élément "hiérarchie" ou "niveau". Dans le cas d'une hiérarchie, l'ordre des membres dans le résultat n'est pas garanti. Si un ordre prévisible est requis, une fonction de classement explicite (telle que hierarchize) doit être utilisée.

### Syntaxe

```
members ( hierarchy|level )
```

### Exemple 1

```
members ( [great_outdoors_company].[Years].[Years] )
```

Résultat

Renvoie les membres contenus dans Années.

### Exemple 2

```
members ( [great_outdoors_company].[Products].[Products].[Product  
line] )
```

Résultat

Renvoie les membres contenus dans Ligne de produits.

## nextMember

Renvoie le membre suivant du niveau dans lequel "membre" se trouve.

### Syntaxe

```
nextMember ( member )
```

### Exemple

```
nextMember ( [Outdoor Protection] )
```

Résultat

Matériel de golf

## openingPeriod

Renvoie le premier membre de même niveau parmi les descendants d'un membre situé au niveau "niveau". Cette fonction est généralement utilisée avec une dimension de temps.

### Syntaxe

```
openingPeriod ( level [ , member ] )
```

### Exemple 1

```
openingPeriod ( [great_outdoors_company].[Years].[Years].[Month] )
```

Résultat

2004/Jan

### Exemple 2

```
openingPeriod ( [great_outdoors_company].[Years].[Years].[Year] )
```

Résultat

2004

### Exemple 3

```
openingPeriod ( [great_outdoors_company].[Years].[Years].[Month] ,  
[2006 Q 4] )
```

Résultat

2006/Oct

## order

Dispose les membres de "expression\_ensemble" en fonction de leur valeur "expression\_valeur" et du troisième paramètre. Les options SC et DESC classent les membres par ordre croissant ou décroissant, respectivement d'après leur position dans la hiérarchie de l'ensemble. Ensuite, les enfants de chaque membre sont classés en fonction de la valeur "value\_expression". Les options BASC et BDESC classent les membres d'un ensemble sans tenir compte de la hiérarchie. En l'absence de toute indication explicite, l'option ASC devient la valeur par défaut.

### Syntaxe

```
order ( set_expression , value_expression [ , ASC|DESC|BASC|BDESC ] )
```

### Exemple 1

```
order ( members ( [Great Outdoors Company].[Product].[Product].  
[Product type] ) , [Quantity sold] , BASC )
```

Résultat

Renvoie la quantité vendue pour chaque type de produit sans respecter d'ordre particulier.

Données du résultat

<b>ProdLine</b>	<b>Quantity</b>
Woods	13,924
Irons	14,244
Safety	22,332
...	...
Sun	215,432
Repellent	270,04
Lanterns	345,096

## Exemple 2

```
order ( members ( [Great Outdoors Company].[Product].[Product].
[Product type] ) , [Quantity sold] , ASC )
```

Résultat

Renvoie la quantité vendue pour chaque type de produit par ordre croissant.

Données du résultat

<b>ProdLine</b>	<b>Quantity</b>
Woods	13,924
Irons	14,244
Putters	23,244
...	...
Tents	130,664
Cooking	198,676
Lanterns	345,096

## ordinal

Renvoie la valeur d'ordonnée basée sur zéro (distance à partir du niveau racine) de "niveau".

### Syntaxe

```
ordinal ( level )
```

### Exemple 1

```
ordinal ( [great_outdoors_company].[Products].[Products].[Product
line] )
```

Résultat

1

### Exemple 2

```
ordinal ( [great_outdoors_company].[Products].[Products].[Product
type] )
```

Résultat

2

## parallelPeriod

Renvoie le membre appartenant à une période précédente dont la position relative est la même que celle de "membre". Cette fonction est semblable à la fonction cousin, mais elle est plus appropriée lorsqu'il s'agit de séries temporelles. Prend l'ancêtre de "membre" au niveau "niveau" (appelé ancêtre) et l'élément de même niveau que "ancêtre" qui est décalé de "expression\_entier" positions et renvoie la période parallèle de "membre" parmi les descendants de cet élément de même niveau. Si "expression\_entier" n'est pas défini, il prend la valeur 1 par défaut et "membre" utilise par défaut le membre en cours.

### Syntaxe

```
parallelPeriod ( level [ , integer_expression [ , member ] ] )
```

### Exemple 1

```
parallelPeriod ( [great_outdoors_company].[Years].[Years].[Quarter] ,  
-1 , [2006/Aug] )
```

Résultat

2006/Nov

### Exemple 2

```
parallelPeriod ( [great_outdoors_company].[Years].[Years].[Quarter] ,  
1 , [2006/Aug] )
```

Résultat

2006/Mai

### Exemple 3

```
parallelPeriod ( [great_outdoors_company].[Years].[Years].[Year] , 2 ,  
[2006/Aug] )
```

Résultat

2004/Aou

## parent

Renvoie le membre qui est le parent de "membre" ou de "mesure".

### Syntaxe

```
parent ( member|measure )
```

### Exemple

```
parent ( [Cooking Gear] )
```

Résultat

Matériel de camping

## periodsToDate

Renvoie un ensemble de membres de même niveau que "membre", tel que déterminé par "niveau". Cette fonction trouve l'ancêtre de "membre" au niveau "niveau" et renvoie les descendants de cet ancêtre qui se trouvent au même niveau

que "membre" (jusqu'à et incluant "membre"). Cette fonction est généralement utilisée avec une dimension de temps. Cette fonction apparaît dans l'exemple de rapport interactif Moyennes continues et mobiles.

### Syntaxe

```
periodsToDate ( level , member )
```

### Exemple

```
periodsToDate ( [great_outdoors_company].[Years].[Year] ,  
[2004/Mar] )
```

Résultat

Renvoie les valeurs correspondant à [2004/Jan], [2004/Fév], [2004/Mar]

## prevMember

Renvoie le membre qui précède immédiatement "membre" indiqué sur le même niveau. Cette fonction apparaît dans l'exemple de rapport Croissance des ventes année après année, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
prevMember ( member )
```

### Exemple 1

```
prevMember ( [Outdoor Protection] )
```

Résultat

Accessoires personnels

### Exemple 2

```
prevMember ( [2005] )
```

Résultat

2004

## member

Définit un membre d'après la valeur de "expression\_valeur" dans "hiérarchie". "Chaîne1" identifie le membre créé par cette fonction. Ce membre doit être unique dans la requête et différent des autres membres de la même hiérarchie. La mention "Chaîne2" est utilisée comme légende du membre ; si elle est absente, la légende est vide. Pour garantir des résultats prévisibles, il est recommandé de définir le paramètre "hiérarchie". Remarque : Tous les calculs utilisés comme éléments de groupement dont les éléments de même niveau sont d'autres calculs ou ensembles de membres doivent être affectés de façon explicite à une hiérarchie à l'aide de cette fonction. Dans le cas contraire, les résultats seront imprévisibles. La seule exception étant lorsque le calcul n'implique que des membres d'une même hiérarchie que celle des éléments de même niveau. Dans ce cas, le calcul est considéré comme appartenant à cette hiérarchie.

### Syntaxe

```
member ( value_expression [ , string1 [ , string2 [ ,  
hierarchy ] ] ] )
```

## Exemple

```
member ( total ( currentMeasure within set filter (
    [great_outdoors_company].[Products].[Products].[Product name] ,
    caption ( [great_outdoors_company].[Products].[Products].[Product
name] ) starts with 'B' ) ) , 'BProducts' , 'B Products' ,
    [great_outdoors_company].[Products].[Products] )
```

Résultat

Renvoie la quantité vendue et les revenus pour tous les produits dont le nom commence par la lettre B.

## nestedSet

Cette fonction est uniquement destinée à Analysis Studio.

### Syntaxe

```
nestedSet ( set_expression1 , set_expression2 )
```

## set

Renvoie la liste des membres définis dans l'expression. Les membres doivent appartenir à la même hiérarchie.

### Syntaxe

```
set ( member { , member } )
```

### Exemple

```
set ( [Golf Equipment] , [Irons] , [TrailChef Cup] )
```

Résultat

Renvoie Matériel de golf, Fers et Gobelet ChefDeCamp.

## siblings

Renvoie les enfants du parent du membre spécifié.

### Syntaxe

```
siblings ( member )
```

### Exemple

```
siblings ( [Golf Equipment] )
```

Résultat

Renvoie les éléments apparentés de Matériel de golf.

Données du résultat

```
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```



## tail

Renvoie les "expression\_index" derniers éléments de "expression\_ensemble". La valeur par défaut de "expression\_index" est 1.

### Syntaxe

```
tail ( set_expression [ , index_expression ] )
```

### Exemple 1

```
tail ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) )
```

Résultat

Renvoie le dernier membre de l'ensemble Ligne de produits.

Données du résultat

Personal Accessories

### Exemple 2

```
tail ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) , 2 )
```

Résultat

Renvoie les deux derniers membres de l'ensemble Ligne de produits.

Données du résultat

Outdoor Protection  
Personal Accessories

## topCount

Trie un ensemble en fonction des valeurs de "expression\_numérique" évaluées pour chacun des membres de "expression\_ensemble" et renvoie les "expression\_index" premiers membres.

### Syntaxe

```
topCount ( set_expression , index_expression , numeric_expression )
```

### Exemple

```
topCount ( [great_outdoors_company].[Products].[Products].[Product  
line] , 2 , [Revenue] )
```

Résultat

Renvoie les deux premiers revenus de l'ensemble Ligne de produits.

Données du résultat

<b>Prod line</b>	<b>Revenue</b>
Camping	\$89,713,990.92
Personal	\$31,894,465.86

## topPercent

Trie par ordre décroissant l'ensemble indiqué dans "expression\_ensemble" et renvoie les éléments du niveau le plus élevé à partir de l'ensemble trié dont le pourcentage cumulatif du total est supérieur ou égal à "pourcentage".

### Syntaxe

```
topPercent ( set_expression , percentage , numeric_expression2 )
```

### Exemple

```
topPercent ( set ( [Camping Equipment] , [Golf Equipment] ,  
[Mountaineering Equipment] ) , 40 , [2006] )
```

Résultat

Pour l'ensemble Matériel de camping, Matériel de golf et Matériel de montagne, renvoie les membres ayant le bénéfice brut le plus élevé atteignant un total au moins égal à 40 % du total général pour l'année 2006.

## topSum

Trie par ordre décroissant l'ensemble indiqué dans "expression\_ensemble" et renvoie les éléments du niveau le plus élevé à partir de l'ensemble trié dont le total cumulatif est supérieur ou égal à "valeur".

### Syntaxe

```
topSum ( set_expression , value , numeric_expression2 )
```

### Exemple

```
topSum ( children ( [Products] ) , 16000000 , tuple ( [2006] ,  
[great_outdoors_company].[Measures].[Gross profit] ) )
```

Résultat

Pour les membres de la ligne de produits, renvoie les membres ayant le bénéfice brut le plus élevé atteignant un total au moins égal à 6 000 000 \$ pour l'année 2006.

## tuple

Indique l'emplacement d'une cellule (intersection) en fonction des membres spécifiés provenant tous de dimensions différentes. Cette fonction inclut par défaut le membre en cours provenant de toutes les dimensions qui ne sont pas définies ailleurs dans les arguments. Le membre en cours de toute dimension qui n'est pas définie dans le contexte d'évaluation est considéré comme étant le membre par défaut de cette dimension. La valeur de cette cellule peut être obtenue à l'aide de la fonction "value".

### Syntaxe

```
tuple ( member { , member } )
```

### Exemple

```
tuple ( [Mountaineering Equipment] , [Fax] )
```

Résultat

Renvoie les ventes d'articles Matériel de montagne par télécopie.

## union

Renvoie des données pour "expression\_ensemble1" et "expression\_ensemble2". Le résultat conserve les doublons uniquement le mot clé facultatif "all" est fourni comme troisième argument.

### Syntaxe

```
union ( set_expression1 , set_expression2 [ , all ] )
```

### Exemple 1

```
union ( set ( [Camping Equipment] , [Golf Equipment] ) , set ( [Golf Equipment] , [Mountaineering Equipment] ) )
```

Résultat

Renvoie des données pour les deux ensembles en tant que nouvel ensemble, en n'affichant la colonne Matériel de golf qu'une seule fois.

### Exemple 2

```
union ( set ( [Camping Equipment] , [Golf Equipment] ) , set ( [Golf Equipment] , [Mountaineering Equipment] ) , all )
```

Résultat

Renvoie des données pour les deux ensembles en tant que nouvel ensemble, en affichant la colonne Matériel de golf deux fois.

## roleValue

Renvoie la valeur de l'attribut qui est associé au rôle dont le nom est indiqué par l'élément "string" dans le contexte spécifié. Le "membre" ou "expression\_ensemble" est facultatif dans un nombre limité de circonstances uniquement, où il peut être dérivé d'un autre contexte. Les applications peuvent être rendues portables pour différentes sources de données et différents modèles en accédant aux attributs par rôle, plutôt que par identificateur d'élément de requête. Pour les sources de données relationnelles modélisées de façon dimensionnelle, l'attribution des rôles incombe au modélisateur. Les rôles prédéfinis des membres de tous les types de sources de données sont les suivants : '\_businessKey', '\_memberCaption', '\_memberDescription', '\_memberUniqueName'. D'autres rôles peuvent être définis dans Framework Manager pour chaque niveau d'une hiérarchie. Par exemple, un niveau Type de produit peut avoir une colonne d'attribut appelée "Conteneur d'expédition type", tandis que le niveau Produit peut posséder un attribut "Conteneur d'expédition de produit". Un rôle personnalisé appelé "Conteneur" peut être affecté à chacun de ces éléments dans Framework Manager. La propriété peut alors être référencée indépendamment du nom de colonne réel, à l'aide de la fonction roleValue.

### Syntaxe

```
roleValue ( string [ , member|set_expression ] )
```

### Exemple 1

```
roleValue ( ' memberCaption' , [Sales].[Product].[Product].[Product line] -> [all].[1] )
```

Résultat

Matériel de camping

## Exemple 2

```
roleValue ( '_businessKey' , [great_outdoors_company].[Years].[Years].  
[Year] )
```

Résultat

Renvoie la valeur de l'attribut qui est associé au rôle `_businessKey`.

Données du résultat

```
("2004-01-01", "2004-12-31")  
("2005-01-01", "2005-12-31")  
("2006-01-01", "2006-12-31")
```

## Exemple 3

```
roleValue ( ' _memberUniqueName' , [great_outdoors_company].[Years].  
[Years].[Year] )
```

Résultat

Renvoie la valeur de l'attribut qui est associé au rôle `_memberUniqueName`.

Données du résultat

```
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (   
Root)].[20040101-20041231]  
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (   
Root)].[20050101-20051231]  
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (   
Root)].[20060101-20061231]
```

## rootMember

Renvoie le membre racine d'une hiérarchie dotée d'une seule racine. Cette fonction apparaît dans l'exemple de rapport Succès des promotions, accessible dans le module Entrepôt de données VA (analyse).

### Syntaxe

```
rootMember ( hierarchy )
```

## rootMembers

Renvoie les membres racine d'une hiérarchie.

### Syntaxe

```
rootMembers ( hierarchy )
```

### Exemple

```
rootMembers ( [great_outdoors_company].[Years].[Years] )
```

Résultat

Par date

## subset

Renvoie un sous-ensemble des membres de "expression\_ensemble" en commençant à "expression\_index1" à partir du début. Si le nombre "expression\_index2" est spécifié, est indiqué, il s'agit du nombre de membres qui seront renvoyés (si disponible). Sinon, tous les membres restants sont renvoyés.

## Syntaxe

```
subset ( set_expression, index_expression1 [ , index_expression2 ] )
```

### Exemple 1

```
subset ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) , 2 )
```

Résultat

Renvoie les membres de l'ensemble Product line à partir du deuxième membre.

Données du résultat

```
Mountaineering Equipment  
Outdoor Protection  
Personal Accessories
```

### Exemple 2

```
subset ( members ( [great_outdoors_company].[Products].[Products].  
[Product line] ) , 2 , 2 )
```

Résultat

Renvoie les deux membres de l'ensemble Product line à partir du deuxième membre.

Données du résultat

```
Mountaineering Equipment  
Outdoor Protection
```

## uniques

Supprime tous les doublons de "expression\_ensemble". Les membres restants conservent leur ordre d'origine.

### Syntaxe

```
unique ( set_expression )
```

## value

Renvoie la valeur de la cellule identifiée par "tuple". Notez que le membre par défaut de la dimension Mesures est Mesure par défaut.

### Syntaxe

```
value ( tuple )
```

### Exemple 1

```
value ( tuple ( [great_outdoors_company].[Years].[Years].[Year]  
->:[PC].[Years (Root)].[20040101-20041231] , [great_outdoors_company].  
[Measures].[Revenue] ) )
```

Résultat

```
34 750 563,50 $
```

## Exemple 2

```
value ( tuple ( [2004] , [Camping Equipment] , [Revenue] ) )
```

Résultat

20 471 328,88 \$

---

## Fonctions de rapport

### Today

Renvoie la date en cours du système.

#### Syntaxe

Today ()

### Now

Renvoie l'heure en cours du système.

#### Syntaxe

Now ()

### AsOfDate

Renvoie la valeur de date de l'expression AsOfDate, si elle est définie. Sinon, AsOfDate renvoie la date d'exécution du rapport.

#### Syntaxe

AsOfDate ()

### AsOfTime

Renvoie la valeur d'heure de l'expression AsOfTime, si elle est définie. Sinon, AsOfTime renvoie l'heure d'exécution du rapport.

#### Syntaxe

AsOfTime ()

### ReportDate

Renvoie la date et l'heure d'exécution du rapport.

#### Syntaxe

ReportDate ()

### ReportName

Renvoie le nom du rapport. Cette fonction est disponible uniquement lorsque le rapport est exécuté à partir IBM® Cognos® Connection. Si une vue de rapport enregistré est exécutée, renvoie le nom de la vue de rapport.

#### Syntaxe

ReportName ()

## ReportPath

Renvoie le chemin du rapport. Cette fonction est disponible uniquement lorsque le rapport est exécuté à partir IBM® Cognos® Connection.

### Syntaxe

ReportPath ()

## ReportDescription

Renvoie la description du rapport. Cette fonction est disponible uniquement lorsque le rapport est exécuté à partir IBM® Cognos® Connection.

### Syntaxe

ReportDescription ()

## ReportLocale

Renvoie les paramètres régionaux d'exécution.

### Syntaxe

ReportLocale ()

## GetLocale

Renvoie les paramètres régionaux d'exécution (obsolète).

### Syntaxe

GetLocale ()

## Locale

Renvoie les paramètres régionaux d'exécution.

### Syntaxe

Locale ()

## ReportProductLocale

Renvoie les paramètres régionaux du produit.

### Syntaxe

ReportProductLocale ()

## ReportAuthorLocale

Renvoie les paramètres régionaux de l'auteur.

### Syntaxe

ReportAuthorLocale ()

## ReportSaveDate

Renvoie la date à laquelle le rapport a été enregistré pour la dernière fois.

### Syntaxe

ReportSaveDate ()

## ReportCreateDate

Renvoie la date à laquelle le rapport a été créé.

### Syntaxe

ReportCreateDate ()

## ReportID

Renvoie l'identificateur de rapport.

### Syntaxe

ReportID ()

## ReportOutput

Renvoie le nom du format de sortie, tel que CSV, HTML, HTMLFragment, layoutDataXML, MHT, PDF, rawXML, spreadsheetML (format Excel 2007), XHTML, xlsxDATA (format de données Excel 2007), XLWA (format Excel 2002), XML, singleXLS (obsolète), XLS (obsolète).

### Syntaxe

ReportOutput ()

## ReportOption

Renvoie la valeur de la variable d'option d'exécution identifiée par "optionName", telle que : attachmentEncoding, burst, cssURL, email, emailAsAttachment, emailAsURL, emailBody, emailSubject, emailTo, emailToAddress, metadataModel, outputEncapsulation, outputFormat, outputLocale, outputPageDefinition, outputPageOrientation, primaryWaitThreshold, print, printer, printerAddress, prompt, promptFormat, saveAs, saveOutput, secondaryWaitThreshold, verticalElements ou xslURL.

### Syntaxe

ReportOption ('optionName')

## ServerName

Renvoie le nom du serveur Web à partir duquel la demande d'exécution a été émise. La valeur peut être vide si la demande est exécutée à partir du programmeur.

### Syntaxe

ServerName ()

## ServerLocale

Renvoie les paramètres régionaux du serveur qui exécute le rapport.

### Syntaxe

ServerLocale ()

## ModelPath

Renvoie le chemin de modèle.



## **Syntaxe**

ModelPath ()

## **BurstKey**

Renvoie la clé de diffusion en rafale.

## **Syntaxe**

BurstKey ()

## **BurstRecipients**

Renvoie la liste de distribution des destinataires de rapports en rafale.

## **Syntaxe**

BurstRecipients ()

## **IsBursting**

Renvoie la valeur booléenne 1 (True) lorsque le rapport est destiné à être diffusé au destinataire ; sinon, renvoie la valeur booléenne 0 (False).

## **Syntaxe**

IsBursting ('recipientName')

## **ParamNames**

Renvoie tous les noms de paramètre.

## **Syntaxe**

ParamNames ()

## **ParamName**

Renvoie le nom de paramètre de la variable "parameterName".

## **Syntaxe**

ParamName ('parameterName')

## **ParamDisplayValue**

Renvoie une chaîne qui correspond à la valeur d'affichage du paramètre "parameterName". Cette fonction apparaît dans l'exemple de rapport Recrutement, accessible dans le module Entrepôt de données VA (analyse).

## **Syntaxe**

ParamDisplayValue ('parameterName')

## **ParamValue**

Renvoie la valeur de paramètre de la variable "parameterName".

## **Syntaxe**

ParamValue ('parameterName')

## **ParamCount**

Renvoie le nombre de paramètres de la variable "parameterName".

## Syntaxe

ParamCount ('parameterName')

## RowNumber

Renvoie la ligne en cours.

## Syntaxe

RowNumber ()

## PageNumber

Renvoie le numéro de la page en cours.

## Syntaxe

PageNumber ()

## PageCount

Renvoie le nombre de pages en cours. Cette fonction est uniquement disponible lorsque la sortie du rapport est de type Adobe® PDF ou Microsoft® Excel. Si vous enregistrez la sortie du rapport, cette fonction est disponible pour tous les formats.

## Syntaxe

PageCount ()

## IsPageCountAvailable

Renvoie la valeur booléenne 1 (True) si le nombre de pages est disponible pour l'exécution actuelle du rapport ; sinon, renvoie la valeur booléenne 0 (False).

## Syntaxe

IsPageCountAvailable ()

## HorizontalPageNumber

Renvoie le numéro de la page horizontale en cours.

## Syntaxe

HorizontalPageNumber ()

## HorizontalPageCount

Renvoie le nombre en cours de pages horizontales.

## Syntaxe

HorizontalPageCount ()

## PageName

Renvoie le nom de la page en cours.

## Syntaxe

PageName ()

## URLEncode

Renvoie la valeur encodée de l'adresse URL du texte d'entrée.

## Syntaxe

`URLEncode ('text')`

## TOCHeadingCount

Renvoie le nombre d'en-têtes de table des matières pour un niveau d'en-tête défini.

## Syntaxe

`TOCHeadingCount ( headingLevel )`

## IsAccessible

Renvoie la valeur booléenne 1 (True) si le rapport est exécuté avec les fonctions d'accessibilité activées. Utilisez cette fonction sous la forme d'une expression de variable avec un bloc conditionnel pour rendre vos rapports accessibles. Par exemple, vous pouvez ajouter une liste ou un tableau croisé équivalent à un graphique dans des rapports exécutés avec les fonctions d'accessibilité activées.

## Syntaxe

`IsAccessible()`

## IsBooklet

Renvoie la valeur booléenne 1 (True) lorsque le rapport est l'enfant d'un livret ; sinon, renvoie la valeur booléenne 0 (False).

## Syntaxe

`IsBooklet()`

## ColumnNumber

Renvoie le numéro de colonne en cours.

## Syntaxe

`ColumnNumber ()`

## IsCrosstabRowNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre de noeud de ligne de tableau croisé.

## Syntaxe

`IsCrosstabRowNodeMember ()`

## IsCrosstabColumnNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre de noeud de colonne de tableau croisé.

## Syntaxe

`IsCrosstabColumnNodeMember ()`

## IsInnerMostCrosstabRowNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre interne de noeud de ligne de tableau croisé.

## Syntaxe

`IsInnerMostCrosstabRowNodeMember ()`

## IsInnerMostCrosstabColumnNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre interne de noeud de colonne de tableau croisé.

## Syntaxe

`IsInnerMostCrosstabColumnNodeMember ()`

## IsOuterMostCrosstabRowNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre externe de noeud de ligne de tableau croisé.

## Syntaxe

`IsOuterMostCrosstabRowNodeMember ()`

## IsOuterMostCrosstabColumnNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est un membre externe de noeud de colonne de tableau croisé.

## Syntaxe

`IsOuterMostCrosstabColumnNodeMember ()`

## IsFirstColumn

Renvoie la valeur booléenne 1 (True) si la colonne en cours est la première colonne.

## Syntaxe

`IsFirstColumn ()`

## IsLastColumn

Renvoie la valeur booléenne 1 (True) si la colonne en cours est la dernière colonne.

## Syntaxe

`IsLastColumn ()`

## IsLastInnerMostCrosstabColumnNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est le dernier membre interne de noeud de colonne de tableau croisé.

## Syntaxe

`IsLastInnerMostCrosstabColumnNodeMember ()`

## IsLastInnerMostCrosstabRowNodeMember

Renvoie la valeur booléenne 1 (True) si le noeud en cours est le dernier membre interne de noeud de ligne de tableau croisé.

## Syntaxe

`IsLastInnerMostCrosstabRowNodeMember ()`

## CubeName

Renvoie le nom du cube. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

### Syntaxe

CubeName ( dimension )

## CubeDescription

Renvoie la description du cube. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

### Syntaxe

CubeDescription ( dimension )

## CubeCreatedOn

Renvoie la date et l'heure auxquelles le cube a été créé. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites. Si la source de dimension est un fichier IBM® Cognos® PowerCube (.mdc), la fonction renvoie une chaîne vide car la date de création initiale d'un PowerCube n'est pas conservée.

### Syntaxe

CubeCreatedOn ( dimension )

## CubeDataUpdatedOn

Renvoie la date et l'heure auxquelles les données du cube ont été mises à jour pour la dernière fois. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

### Syntaxe

CubeDataUpdatedOn ( dimension )

## CubeSchemaUpdatedOn

Renvoie la date et l'heure auxquelles le schéma du cube a été mis à jour pour la dernière fois. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

### Syntaxe

CubeSchemaUpdatedOn ( dimension )

## CubelsOptimized

Renvoie la valeur "true" si le cube est optimisé. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

### Syntaxe

CubeIsOptimized ( dimension )

## CubeDefaultMeasure

Renvoie le nom de la mesure par défaut du cube. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

## Syntaxe

CubeDefaultMeasure ( dimension )

## CubeCurrentPeriod

Renvoie la période en cours du cube. Le paramètre dimension indique à partir de quel cube les métadonnées doivent être extraites.

## Syntaxe

CubeCurrentPeriod ( dimension )

## CellValue

Renvoie la valeur de la cellule actuelle du tableau croisé.

## Syntaxe

CellValue ( )

## InScope

Renvoie la valeur booléenne 1 (True) lorsque la cellule se trouve dans la portée des données élémentaires et des noms uniques de membre ; sinon, renvoie la valeur booléenne 0 (False).

## Syntaxe

InScope ( dataItem , MUN, ... )

---

## Fonctions de conversion du type de données

### \_add\_days

Renvoie l'élément date-heure résultant de l'ajout de "expression\_entier" jours à "expression\_horodatage".

## Syntaxe

\_add\_days ( timestamp\_expression , integer\_expression )

## Exemple

\_add\_days ( 2007-01-14 00:00:00.000 , 3 )

Résultat

2007-01-17 00:00:00.000

### \_add\_months

Renvoie l'élément date-heure résultant de l'ajout de "expression\_entier" mois à "expression\_horodatage".

## Syntaxe

\_add\_months ( timestamp\_expression , integer\_expression )

### \_add\_years

Renvoie l'élément date-heure résultant de l'ajout de "expression\_entier" années à "expression\_horodatage".

## Syntaxe

```
_add_years ( timestamp_expression , integer_expression )
```

## \_age

Renvoie un nombre en soustrayant "expression\_horodatage" à la date du jour.

## Syntaxe

```
_age ( timestamp_expression )
```

## Exemple

```
_age ([Query1].[Date]), where [Query1].[Date] is March 2, 2004, and  
today is July 8, 2009
```

Résultat

50406, où 5 représente le nombre d'années, 04 le nombre de mois et 06 le nombre de jours.

## \_day\_of\_week

Renvoie le jour de la semaine (entre 1 et 7) pour "expression\_horodatage" où "expression\_entier" indique quel jour de cette semaine est le jour 1. Pour déterminer "expression\_entier", choisissez le jour de la semaine et comptez à partir de lundi ; par exemple, si vous choisissez mercredi, "expression\_entier" sera 3 car mercredi est le troisième jour à partir de lundi.

## Syntaxe

```
_day_of_week ( timestamp_expression , integer_expression )
```

## Exemple

```
_day_of_week ( 2009-01-01 , 7 ), where 7 means that Sunday is the  
first day of the week.
```

Résultat

5

## \_day\_of\_year

Renvoie l'ordonnée correspondant au jour de l'année dans "expression\_horodatage" (compris entre 1 et 366). Également appelé jour julien.

## Syntaxe

```
_day_of_year ( timestamp_expression )
```

## \_days\_between

Renvoie un nombre positif ou négatif représentant le nombre de jours entre "expression1\_horodatage" et "expression2\_horodatage". Si "expression1\_horodatage" < "expression2\_horodatage", le résultat est un nombre négatif.

## Syntaxe

```
_days_between ( timestamp_expression1 , timestamp_expression2 )
```

## **\_days\_to\_end\_of\_month**

Renvoie un nombre représentant le nombre de jours restant dans le mois représenté par "expression\_horodatage".

### **Syntaxe**

```
_days_to_end_of_month ( timestamp_expression )
```

## **\_first\_of\_month**

Renvoie une date-heure correspondant au premier jour du mois représenté par "expression\_horodatage".

### **Syntaxe**

```
_first_of_month ( timestamp_expression )
```

### **Exemple 1**

```
_first_of_month ( 2009-05-04 00:00:00.000 )
```

Résultat

Renvoie 2009-05-01 00:00:00.000

### **Exemple 2**

```
_first_of_month ( current_date )
```

Résultat

Renvoie 1 juillet 2009 si la date en cours est 30 juillet 2009.

## **\_last\_of\_month**

Renvoie une date-heure correspondant au dernier jour du mois représenté par "expression\_horodatage".

### **Syntaxe**

```
_last_of_month ( timestamp_expression )
```

## **\_make\_timestamp**

Renvoie un horodatage construit à partir des expressions "expression\_entier1" (année), "expression\_entier2" (mois) et "expression\_entier3" (jour). La valeur par défaut de la portion heure est 00:00:00.000.

### **Syntaxe**

```
_make_timestamp ( integer_expression1 , integer_expression2 ,  
integer_expression3 )
```

## **\_months\_between**

Renvoie un nombre positif ou négatif représentant le nombre de mois entre "expression1\_horodatage" et "expression2\_horodatage". Si "expression1\_horodatage" < "expression2\_horodatage", le résultat est un nombre négatif.

### **Syntaxe**

```
_months_between ( timestamp_expression1 , timestamp_expression2 )
```



## **\_week\_of\_year**

Renvoie le numéro de la semaine dans l'année (entre 1 et 53) pour "expression\_horodatage". Selon la norme ISO 8601, la semaine 1 correspond à la première semaine de l'année qui compte un jeudi, ce qui correspond à la semaine du 4 janvier. Une semaine commence le lundi (jour 1) et se termine le dimanche (jour 7).

### **Syntaxe**

```
_week_of_year ( timestamp_expression )
```

## **\_years\_between**

Renvoie un nombre entier positif ou négatif représentant le nombre d'années entre "expression1\_horodatage" et "expression2\_horodatage". Si "expression1\_horodatage" < "expression2\_horodatage", la valeur renvoyée est négative.

### **Syntaxe**

```
_years_between ( timestamp_expression1 , timestamp_expression2 )
```

## **\_ymdint\_between**

Renvoie un nombre représentant la différence entre "expression1\_horodatage" et "expression2\_horodatage". Cette valeur prend la forme AAMMJJ, où AA représente le nombre d'années, MM le nombre de mois et JJ le nombre de jours.

### **Syntaxe**

```
_ymdint_between ( timestamp_expression1 , timestamp_expression2 )
```

### **Exemple**

```
_ymdint_between ( [Query1].[Date (close date)] , [Query1].[Date (ship date)] ), where [Query1].[Date (close date)] is February 20, 2004, and [Query1].[Date (ship date)] is January 19, 2004.
```

Résultat

101, où 1 est le nombre de mois et 01 est le nombre de jours.

## **abs**

Renvoie la valeur absolue de "expression\_numérique". Si "expression\_numérique" est une valeur négative, la valeur renvoyée est positive.

### **Syntaxe**

```
abs ( numeric_expression )
```

## **partie entière par excès**

Renvoie le plus petit entier supérieur ou égal à "expression\_numérique".

### **Syntaxe**

```
ceiling ( numeric_expression )
```

## **character\_length**

Renvoie le nombre de caractères contenus dans "expression\_chaine".

## Syntaxe

`character_length ( string_expression )`

## date2string

Renvoie une date sous forme de chaîne au format AAAA-MM-JJ.

## Syntaxe

`date2string ( date_expression )`

## date2timestamp

Convertit "expression\_date" en horodatage. La partie heure de l'horodatage sera égale à zéro.

## Syntaxe

`date2timestamp ( date_expression )`

## date2timestampTZ

Convertit une "expression\_date" en horodatage avec fuseau horaire. Les parties heure et fuseau horaire de l'horodatage seront égales à zéro.

## Syntaxe

`date2timestampTZ ( date_expression )`

## DTinterval2string

Renvoie un intervalle date-heure sous forme de chaîne au format JJJJ HH:MM:SS.FFFFFFFF ou -JJJJ HH:MM:SS.FFF.

## Syntaxe

`DTinterval2string ( date_time_interval_expression )`

## DTinterval2stringAsTime

Renvoie un intervalle date-heure sous forme de chaîne au format HHHH:MM:SS.FFFFFFFF ou HH:MM:SS.FFF. Les jours sont convertis en heures.

## Syntaxe

`DTinterval2stringAsTime ( date_time_interval_expression )`

## exp

Renvoie la constante "e" à la puissance "expression\_numérique". La constante "e" est la base du logarithme naturel.

## Syntaxe

`exp ( numeric_expression )`

## Exemple

`exp ( 2 )`

Résultat

7,389056

## extract

Renvoie un entier représentant la valeur de "expression\_partie\_date" en "expression\_date\_heure". L'expression "expression\_partie\_date" peut être l'année, le mois, le jour, l'heure, la minute ou la seconde.

### Syntaxe

```
extract ( date_part_expression , datetime_expression )
```

### Exemple 1

```
extract ( year , 2003-03-03 16:40:15.535 )
```

Résultat

2003

### Exemple 2

```
extract ( hour , 2003-03-03 16:40:15.535 )
```

Résultat

16

## partie-entière

Renvoie le plus grand entier inférieur ou égal à "expression\_numérique".

### Syntaxe

```
floor ( numeric_expression )
```

## int2DTinterval

Convertit un entier en un intervalle date-heure. "expression\_chaine" précise ce que "expression\_entier" représente : "ns" = nanosecondes, "s" = secondes (valeur par défaut), "m" = minutes, "h" = heures, "j" = jours.

### Syntaxe

```
int2DTinterval ( integer_expression , string_expression )
```

### Exemple 1

```
int2DTinterval (1020,"h")
```

Résultat

42 jours et 12 heures

### Exemple 2

```
int2DTinterval (1020,"s")
```

Résultat

17 minutes

## int2YMinterval

Convertit "expression\_entier" en un intervalle année-mois. "expression\_chaine" précise ce que "expression\_entier" représente : "a" = années, "m" = mois (valeur par défaut).

### Syntaxe

```
int2YMinterval ( integer_expression , string_expression )
```

## ln

Renvoie le logarithme naturel de "expression\_numérique".

### Syntaxe

```
ln ( numeric_expression )
```

## minuscules

Convertit en minuscules tous les caractères en majuscules de "expression\_chaine". Cette fonction apparaît dans l'exemple de rapport Performances de ventes diffusées en rafale, accessible dans le module Entrepôt de données VA (requête).

### Syntaxe

```
lower ( string_expression )
```

## mapNumberToLetter

Ajoute "expression\_entier" à "expression\_chaine".

### Syntaxe

```
mapNumberToLetter ( string_expression , integer_expression )
```

### Exemple

```
mapNumberToLetter ( 'a' , 1 )
```

Résultat

b

## mod

Renvoie une valeur entière représentant le reste (modulo) de "expression1\_entier"/"expression2\_entier".

### Syntaxe

```
mod ( integer_expression1 , integer_expression2 )
```

## nullif

Renvoie la valeur NULL si "expression1\_chaine" et "expression2\_chaine" sont identiques (sans respecter la casse), sinon renvoie "expression1\_chaine".

### Syntaxe

```
nullif ( string_expression1 , string_expression2 )
```

## number2string

Convertit une "expression\_numérique" en chaîne, à l'aide de l'indicateur de format %g (C/C++ syntax).

### Syntaxe

```
number2string ( numeric_expression )
```

## octet\_length

Renvoie le nombre d'octets contenus dans "expression\_chaine".

### Syntaxe

```
octet_length ( string_expression )
```

## position

Renvoie l'entier représentant la position de début de "expression1\_chaine" dans "expression2\_chaine". Renvoie 0 si "expression\_chaine1" est introuvable.

### Syntaxe

```
position ( string_expression1 , string_expression2 )
```

## power

Renvoie "expression1\_numérique" à la puissance "expression2\_numérique".

### Syntaxe

```
power ( numeric_expression1 , numeric_expression2 )
```

### Exemple

```
power ( 3 , 2 )
```

Résultat

9

## round

Renvoie l'expression "expression\_numérique" arrondie à la valeur la plus proche avec "expression\_entier" chiffres significatifs à droite du séparateur décimal. Si "expression\_entier" est une valeur négative, "expression\_numérique" est arrondi à la valeur absolue la plus proche avec "expression\_entier" chiffres significatifs à gauche du séparateur décimal. L'arrondi est calculé avant que le formatage des données ne soit appliqué.

### Syntaxe

```
round ( numeric_expression , integer_expression )
```

### Exemple

```
round (125, -1)
```

Résultat

130

## racine-carrée

Renvoie la racine carrée de "expression\_numérique". La valeur "expression\_numérique" ne doit pas être négative.

### Syntaxe

```
sqrt ( numeric_expression )
```

## status

Renvoie le statut de l'"expression". Les valeurs possibles sont : 0 - OK, 1 - null, 2 - indisponible, 4 - division par zéro, 8 - dépassement, 16 - sécurité, 32 - erreur, 64 - nouveau, 128 - échantillon, 256 - en attente.

### Syntaxe

```
status ( expression )
```

## string2date

Retourne une "expression\_chaîne" sous forme de date au format AAAA-MM-JJ.

### Syntaxe

```
string2date ( string_expression )
```

## string2double

Renvoie un nombre à virgule flottante. "expression\_chaîne" se présente sous la forme suivante : "[blanc] [signe] [chiffres] [chiffres] [ {d|D|e|E }][signe]chiffres]"

### Syntaxe

```
string2double ( string_expression )
```

## string2DTinterval

Retourne une "expression\_chaîne" sous forme d'intervalle date-heure au format [-]JJ HH:MM[:SS[.FFF]].

### Syntaxe

```
string2DTinterval ( string_expression )
```

## string2int32

Renvoie un entier. Le format de "expression\_chaîne" est le suivant : "[espace] [[+|-]] [chiffres]"

### Syntaxe

```
string2int32 ( string_expression )
```

## string2int64

Renvoie une chaîne de type entier long. Le format de "expression\_chaîne" est le suivant : "[espace] [[+|-]] [chiffres]"

### Syntaxe

```
string2int64 ( string_expression )
```

## string2time

Renvoie une "expression\_chaîne" sous forme d'heure au format HH:MM:SS.FFFFFFFF.

### Syntaxe

```
string2time ( string_expression )
```

## string2timestamp

Renvoie une "expression\_chaîne" sous forme d'horodatage au format AAAA-MM-JJ [T|t|[blanc]+] HH:MM:SS.FFFFFFFF.

### Syntaxe

```
string2timestamp ( string_expression )
```

## string2timestampTZ

Renvoie une "expression\_chaîne" au format AAAA-MM-JJ HH:MM:SS.FFFFFFFF +HHMM ou AAAA-MM-JJ [T|t] HH:MM:SS.FFF -HHMM.

### Syntaxe

```
string2timestampTZ ( string_expression )
```

## string2YMinterval

Retourne une "expression\_chaîne" sous forme d'intervalle année-mois au format [-]AA MM.

### Syntaxe

```
string2YMinterval ( string_expression )
```

## sous-chaîne

Renvoie la sous-chaîne de "expression\_chaîne" qui commence à la position "expression1\_entier" pour "expression2\_entier" caractères ou à la fin de "expression\_chaîne" si "expression2\_entier" est égal à -1. Le premier caractère de "expression\_chaîne" occupe la position 1.

### Syntaxe

```
substring ( string_expression , integer_expression1 ,  
integer_expression2 )
```

### Exemple

```
substring ( [Sales (analysis)].[Sales staff].[Sales staff].[Sales  
staff].[Position code], 3 , 5 )
```

Résultat

Renvoie les caractères 3 à 7 des codes de position.

## time2string

Renvoie une heure sous forme de chaîne au format HH:MM:SS.FFF.

### Syntaxe

```
time2string ( time_expression )
```

## timestamp2date

Convertit "expression\_horodatage" en date. La partie heure de l'horodatage sera ignorée.

### Syntaxe

```
timestamp2date ( timestamp_expression )
```

## timestamp2string

Renvoie un horodatage sous forme de chaîne au format "AAAA-MM-JJ HH:MM:SS.FFFFFFFF".

### Syntaxe

```
timestamp2string ( timestamp_expression )
```

## timestamp2timestampTZ

Convertit une "expression\_horodatage" en horodatage avec un fuseau horaire. La partie décalage de l'horodatage avec fuseau horaire affichera zéro.

### Syntaxe

```
timestamp2timestampTZ ( timestamp_expression )
```

## timestampTZ2date

Convertit "expression\_horodatage\_fuseau\_horaire" en date. Les parties heure et fuseau horaire de l'horodatage seront ignorées.

### Syntaxe

```
timestampTZ2date ( timestamp_time_zone_expression )
```

## timestampTZ2string

Renvoie un horodatage avec fuseau horaire sous forme de chaîne au format AAAA-MM-JJ HH:MM:SS.FFFFFFFF +HHMM ou AAAA-MM-JJ HH:MM:SS.FFF -HHMM.

### Syntaxe

```
timestampTZ2string ( timestamp_time_zone_expression )
```

## timestampTZ2timestamp

Convertit "expression\_horodatage\_fuseau\_horaire" en horodatage. La partie décalage de l'horodatage avec fuseau horaire sera ignorée.

### Syntaxe

```
timestampTZ2timestamp ( timestamp_time_zone_expression )
```

## timeTZ2string

Renvoie une heure avec un fuseau horaire sous forme de chaîne au format HH:MM:SS.FFF +HHMM" ou "HH:MM:SS.FFFFFFFF -HHMM. Par exemple, -05:30 indique qu'il s'agit du fuseau horaire GMT moins 5 heures et 30 minutes.

### Syntaxe

```
timeTZ2string ( timeTZ_expression )
```



## **trim**

Renvoie une expression "expression\_chaine" sans les blancs de droite ou de gauche ou sans le caractère indiqué dans "expression\_caractère\_correspondance". "expression\_blanc" peut être : "LEADING", "TRAILING" ou "BOTH" (valeur par défaut). "expression\_caractère\_correspondance" peut être une chaîne vide pour supprimer les blancs ; sinon, spécifiez le caractère à supprimer.

### **Syntaxe**

```
trim ( trim_what_expression , match_character_expression ,  
      string_expression )
```

## **majuscules**

Convertit tous les caractères minuscules de "expression\_chaine" en majuscules.

### **Syntaxe**

```
upper ( string_expression )
```

## **YMinterval2string**

Renvoie "expression\_intervalle\_année\_mois" sous forme de chaîne au format (AA MM) ou -(AA MM).

### **Syntaxe**

```
YMinterval2string ( year_month_interval_expression )
```



---

## Annexe D. Aide-mémoire sur le formatage de données

Ce chapitre contient des définitions se rapportant aux propriétés de formatage de données disponibles dans IBM Cognos Framework Manager. La définition de chaque propriété de formatage est également affichée lorsque vous sélectionnez une propriété dans la boîte de dialogue **Format des données** dans Framework Manager.

Pour les métadonnées SAP BW :

- Toute unité d'informations de mesure qui existe dans SAP BW est automatiquement ajoutée à la valeur de données.
- Vous ne pouvez pas définir un format pour chaque devise dans un sujet de requête multidevise.

---

### Propriétés de formatage de données

La liste qui suit répertorie les propriétés disponibles dans la boîte de dialogue de formatage des données.

#### Caractères 'non applicable'

Indique les caractères à afficher lorsque la valeur à formater n'est pas applicable. La valeur par défaut est deux tirets (--). Notez que le format sera appliqué seulement si la source de données peut prendre en charge cette condition d'erreur.

#### Caractères d'erreur (Tous)

Indique les caractères à afficher lorsque la valeur à formater n'est pas disponible en raison d'une erreur. Cette propriété est remplacée par des conditions d'erreurs plus spécifiques comme les caractères d'erreur de sécurité. La valeur par défaut est deux tirets (--). Notez que le format sera appliqué seulement si la source de données peut prendre en charge cette condition d'erreur.

#### Type de calendrier

Indique le type de calendrier à afficher. Les valeurs de date sont mappées avec le calendrier sélectionné avant d'appliquer la mise en forme. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur. Il est à noter que la valeur Impérial japonais ne s'applique qu'au japonais.

#### Format de l'heure

Indique le format d'affichage de l'heure : 12 heures ou 24 heures. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

#### Devise

Indique quelle devise utiliser. Le symbole de la devise par défaut s'affiche, à moins que des changements soient apportés aux propriétés Afficher la devise et Symbole de la devise. La valeur par défaut est héritée du modèle.

#### Afficher la devise

Indique quel symbole afficher : le symbole international de la devise ou le symbole de la devise locale. La valeur par défaut est le symbole de la devise locale.

## Symbole de la devise

Indique un ou plusieurs caractères à utiliser comme symbole pour désigner la devise locale. Le symbole est inséré avant le nombre et avant tout signe qui le précède. Pour ajouter un espace entre le symbole de la devise et la valeur numérique, il suffit d'indiquer cet espace dans la propriété, après le symbole. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Position du symbole de la devise

Indique le point d'insertion du symbole de la devise. Par exemple, si la valeur Fin est sélectionnée, le symbole (c'est-à-dire les caractères indiqués dans la propriété Symbole de la devise ou Symbole international de la devise) sera inséré après le nombre ; il peut être précédé d'un espace si celui-ci a été défini dans la valeur de la propriété du symbole. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Ordre de la date

Indique l'ordre d'affichage des trois éléments de la date : jour, mois et année. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Séparateur de date

Indique quel caractère afficher entre l'année, le mois et le jour. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Style de date

Indique le style de date. Les résultats sont affichés en fonction de la langue. En règle générale, le format Abrégé n'affiche que les chiffres ; le format Moyen utilise des abréviations et le format Long affiche tous les détails disponibles.

## Séparateur de décimales

Indique quel caractère sépare les décimales du nombre qui les précède. Cette propriété n'a aucun effet si les décimales ne sont pas affichées. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Format des caractères numériques

Définit l'option de format applicable aux caractères numériques. Lorsque l'option National est sélectionnée, le format des caractères numériques est déterminé par la langue du contenu utilisateur. Lorsque l'option Contextuel est sélectionnée, leur format est déterminé par les caractères contigus dans la valeur. Par exemple, si le dernier caractère fortement directionnel avant le caractère numérique est du type de gauche à droite, le nombre s'affiche au format européen. Si le dernier caractère fortement directionnel avant le caractère numérique est du type droite à gauche, le nombre s'affiche au format numérique arabe, variante orientale (Arabic-Indic). S'il n'y a pas de caractère fortement directionnel avant le caractère numérique, l'affichage du nombre dépend de la direction de base de la zone de texte. Avec l'option Aucun, aucune mise en forme n'est réalisée, et la valeur s'affiche de la même façon que dans la source de données. La valeur par défaut dépend du type de valeur. Lorsque la valeur est une chaîne, la valeur par défaut est Aucun.

## Format des caractères numériques

Définit l'option de format applicable aux caractères numériques. Lorsque l'option National est sélectionnée, le format des caractères numériques est déterminé par la langue du contenu utilisateur. Avec l'option Aucun, aucune mise en forme n'est

réalisée, et la valeur s'affiche de la même façon que dans la source de données. La valeur par défaut dépend du type de valeur. Lorsque la valeur est un nombre, la valeur par défaut est le format ICU par défaut de la langue sélectionnée.

### **Afficher les symboles AM / PM**

Indique si les symboles AM et PM seront affichés. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher sous forme d'exposant**

Indique si les valeurs doivent être affichées selon la notation décimale, à l'aide d'exposants. Si la valeur de cette propriété est Non, aucune notation scientifique ne sera utilisée. Si aucune valeur n'est indiquée pour cette propriété, la notation scientifique ne sera utilisée que dans les cas où la valeur excède le nombre maximal de chiffres autorisé. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les jours**

Indique s'il faut afficher le jour. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. La valeur Julien signifie que les trois chiffres représentant le jour de l'année seront affichés. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les périodes**

Indique s'il faut afficher l'ère. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les heures**

Indique s'il faut afficher les heures. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les millisecondes**

Indique s'il faut afficher les millisecondes. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. Cette propriété n'a aucun effet si les secondes ne sont pas affichées. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les minutes**

Indique s'il faut afficher les minutes. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les mois**

Indique s'il faut afficher le mois. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

### **Afficher les mois**

Indique s'il faut afficher le mois. N'utilisez cette propriété que lorsque le rapport est traité entièrement dans la base de données et qu'aucun traitement local n'est requis. Le type d'intervalle des colonnes de base de données associées doit être ANNEE-MOIS, et non JOUR-HEURE.

## Afficher les secondes

Indique s'il faut afficher les secondes. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Afficher les fuseaux horaires

Indique s'il faut afficher le fuseau horaire. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Afficher les jours de la semaine

Indique s'il faut afficher le jour de la semaine. Dans l'affirmative, le format affiché sera celui sélectionné dans la liste déroulante. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Afficher les années

Indique s'il faut afficher l'année. L'affichage des deux premiers chiffres de l'année, qui indique le siècle, est personnalisable; il suffit de sélectionner l'une des valeurs associées à la propriété. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Afficher les années

Indique s'il faut afficher l'année. N'utilisez cette propriété que lorsque le rapport est traité entièrement dans la base de données et qu'aucun traitement local n'est requis. Le type d'intervalle des colonnes de base de données associées doit être ANNEE-MOIS, et non JOUR-HEURE.

## Caractères de division par zéro

Indique les caractères à afficher lorsqu'une valeur numérique est le résultat d'une division par zéro. La valeur par défaut est /0. Notez que le format sera appliqué seulement si la source de données peut prendre en charge cette condition d'erreur.

## Symbole de l'exposant

Indique le caractère d'identification des exposants quand les nombres sont affichés selon la notation scientifique. Ce caractère sera inséré après le nombre et séparé par un espace. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Taille du groupe (chiffres)

Indique la taille du groupe principal. Cette valeur représente le nombre de chiffres regroupés à la gauche du point décimal et divisés par le séparateur des milliers. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Symbole international de la devise

Indique un ou plusieurs caractères à utiliser pour désigner le symbole international de la devise. Ce symbole remplace celui de la devise. Afin d'ajouter un espace entre le symbole de la devise et la valeur numérique, il suffit d'indiquer cet espace dans la propriété, après le symbole. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Mantisse (chiffres)

Indique le nombre de chiffres à afficher après le symbole de l'exposant quand les nombres sont affichés selon la notation scientifique.

## Nb. maximal de chiffres

Indique le nombre maximal de chiffres qui peuvent être affichés. Dans les cas où la valeur à afficher excède ce nombre, la notation scientifique sera utilisée. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Nb. minimal de chiffres

Indique le nombre minimal de chiffres qui peuvent être affichés. Dans les cas où la valeur à afficher est en deçà de ce nombre, le caractère de cadrage sera utilisé. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Caractères pour les valeurs manquantes

Indique le ou les caractères à afficher quand il manque une valeur. Si aucune valeur n'est indiquée pour cette propriété, une chaîne vide sera affichée.

## Motif des nombres négatifs

Indique un format de présentation des nombres négatifs à l'aide de motifs. Certaines restrictions s'appliquent. La partie numérique du motif négatif est ignorée. Le préfixe et le suffixe sont les seuls éléments utilisés. Par exemple, dans le motif 'ABC# ##0,#EFG', 'ABC' est le préfixe, 'EFG' est le suffixe et '# ##0,#' représente la partie numérique du motif.

## Position du signe négatif

Indique le point d'insertion du signe négatif. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Signe négatif

Indique le mode d'affichage des nombres négatifs. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Nombre de décimales

Indique le nombre de décimales à afficher. Si cette propriété n'est pas définie, le nombre de décimales variera en fonction du nombre affiché.

## Caractères de dépassement numérique

Indique les caractères à afficher lorsqu'une valeur numérique est le résultat d'un dépassement numérique. La valeur par défaut est deux tirets (--). Notez que le format sera appliqué seulement si la source de données peut prendre en charge cette condition d'erreur.

## Caractère de cadrage

Indique le caractère qui sert à remplir les valeurs dont le nombre de chiffres est en deçà du nombre minimal indiqué. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Motifs

Indique un format de présentation à l'aide de motifs. Le format du modèle remplace les formats indiqués dans les autres propriétés. A titre d'exemple, pour mettre la date au format 2009/12/31 23:59:59 PM, utilisez le modèle `aaaa/mm/jj hh:mm:ss aa`. Par exemple, pour formater les milliers en utilisant la lettre K, définissez le type de format numérique, définissez l'échelle sur -3 (pour supprimer 000), puis appliquez le modèle `####K`.

## Symbole de pourcentage

Indique si les valeurs seront affichées selon une échelle de pour cent ou de pour mille. Le symbole est inséré après le nombre et tout signe à la droite du nombre. Afin d'ajouter un espace entre le symbole de la devise et la valeur numérique, il suffit d'indiquer cet espace dans la propriété, après le symbole. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Echelle des pourcentages (entier)

Echelle à appliquer à la valeur après le formatage. Si cette propriété n'est pas définie, aucune échelle de pourcentage ne sera appliquée et la valeur sera formatée en fonction du nombre de positions décimales normal associé au symbole de pourcentage (ou pour mille).

## Echelle

Indique, en chiffres, le décalage du signe décimal aux fins de formatage. Par exemple, afin de présenter les valeurs selon l'échelle pour mille, déplacer le signe décimal de trois espaces. La valeur par défaut est héritée de la zone correspondante dans la base de données.

## Taille du groupe secondaire (chiffres)

Indique la taille du groupe secondaire. Cette valeur représente le nombre de chiffres regroupés à la gauche du groupe principal et divisés par le séparateur des milliers. Si aucune valeur n'est indiquée, le nombre de chiffres dans le groupe secondaire est le même que celui du groupe principal, conformément aux informations contenues dans la propriété Taille du groupe (chiffres). La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Caractères d'erreur de sécurité

Indique les caractères à afficher lorsque la valeur à formater n'est pas disponible pour des raisons de sécurité. La valeur par défaut est `#!Security`. Notez que le format sera appliqué seulement si la source de données peut prendre en charge cette condition d'erreur.

## Séparateur des milliers

Indique le mode de séparation des groupes de chiffres, par exemple les milliers. Cette propriété n'est utilisée que si la propriété Séparateur des milliers est définie à Oui. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## Séparateur de l'heure

Indique quel caractère afficher entre l'heure, les minutes et les secondes. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.



## **Format de l'heure**

Indique le format d'affichage de l'heure. Les résultats affichés dépendent de la langue. En règle générale, le format Abrégé n'affiche qu'un minimum de détails; le format Long inclut les secondes et le format Complet affiche tous les détails, y compris le fuseau horaire. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## **Unité de temps**

Indique l'unité de mesure de la valeur. Cette propriété n'a aucun effet si l'une des parties jour ou heure est affichée. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## **Utiliser le séparateur des milliers**

Indique si le séparateur de groupes sera appliqué conformément à la valeur de la propriété Taille du groupe. La valeur par défaut est héritée de la langue de contenu sélectionnée par l'utilisateur.

## **Caractères pour les valeurs nulles**

Indique le ou les caractères à afficher quand la valeur est de zéro (0). Si aucune valeur n'est indiquée pour cette propriété, le nombre de zéros affichés sera basé sur la valeur de la propriété 'Nb. maximal de chiffres'.



---

## Annexe E. Utilisation de motifs pour mettre en forme des données

Il est possible de formater des données pour qu'elles utilisent n'importe quel motif alphanumérique lorsque les formats par défaut ne répondent pas aux besoins. Par exemple, vous pouvez formater les dates pour qu'elles s'inscrivent en lettres, y compris l'ère, ou pour qu'elles s'inscrivent en chiffres et n'affichent que les deux derniers chiffres des années afin d'économiser de l'espace.

L'utilisation de symboles et de motifs peut donner des résultats semblables à ceux obtenus lors de l'utilisation des tâches de formatage de base. Par exemple, il est possible de définir combien de chiffres seront affichés après le séparateur décimal. Vous pouvez obtenir ce genre de résultats avec un motif ou définir la propriété **Nombre de décimales**. Les motifs offrent la flexibilité nécessaire pour répondre à des exigences complexes.

Chacun des codes de langue de contenu pris en charge requiert un ensemble de symboles précis qui seront utilisés dans les motifs. Vous aurez besoin de deux tables pour chacun des codes de langue ; une pour les symboles de date et d'heure et l'autre pour les symboles décimaux. Les symboles décimaux sont identiques pour tous les paramètres régionaux, mais les symboles de date et d'heure sont regroupés en six groupes de paramètres régionaux. Consultez la section relative aux symboles de date et d'heure pour savoir quel groupe de paramètres régionaux est utilisé pour vos paramètres régionaux.

Pour définir des motifs, ouvrez la boîte de dialogue **Format des données** et modifiez la propriété **Motif** pour chaque type de format. Utilisez les symboles qui sont définis dans les tables de codes de langues et suivez les instructions indiquées ci-après.

---

### Règles d'utilisation des motifs

Lorsque vous définissez un motif, le nombre de symboles que vous utilisez a une incidence sur l'affichage des données. Il existe des règles différentes pour le texte et pour les chiffres, ainsi que pour les valeurs qui peuvent être affichées sous forme de texte ou de chiffres.

#### Texte

Vous pouvez indiquer si le texte produit doit être abrégé ou non.

Nombre de symboles	Description	Exemple
4 ou plus	Format long	EEEE correspond à Lundi
Inférieur à 4	Format abrégé	EEE correspond à Lun.

#### Chiffres

Le nombre de symboles que vous utilisez dans un motif définit le nombre minimal de chiffres qui seront affichés dans un rapport. Des zéros sont ajoutés aux nombres

qui comptent moins de chiffres que le nombre défini. Par exemple, si vous définissez mm pour les minutes et que la valeur de la base de données est 6, le rapport affichera 06.

**Remarque :** La valeur de l'année observe des règles différentes. Si vous définissez deux symboles pour l'année, les deux derniers chiffres de la valeur de l'année seront affichés. Par exemple, aaaa renvoie 2013, tandis que aa renvoie 13.

## Texte et chiffres

Pour des valeurs qui sont affichées sous forme de texte ou de chiffres, notamment les mois, vous pouvez décider d'afficher les données sous forme de texte ou de chiffres et, le cas échéant, d'utiliser le format long ou abrégé.

Nombre de symboles	Description	Exemple
3 ou plus	Texte	MMMM renvoie Janvier MMM renvoie Jan.
Inférieur à 3	Chiffres	MM renvoie 01 M renvoie 1

## Symboles de date et d'heure

Les symboles de date et d'heure sont divisés en paramètres régionaux répartis comme suit :

### Groupe de paramètres régionaux A

Paramètres régionaux : af-za, en, en-au, en-be, en-bw, en-ca, en-gb, en-hk, en-ie, en-in, en-mt, en-nz, en-ph, en-sg, en-us, en-vi, en-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, ms-my, nb-no, nl, nl-be, nl-nl, no, no-no, om-et, om-so, pl, pl-pl, pt, pt-br, pt-pt, so-dj, so-et, so-ke, so-so, sv, sv-fi, sv-se, sw-ke, sw-tz

Signification	Symbole	Présentation	Exemple
Ere	G	Texte	AD
Année	a	Numérique	2013
Année (de 'Semaine de l'année')	A	Numérique	2013
Mois de l'année	M	Texte et chiffres	Juillet et 07
Semaine de l'année	w	Numérique	27
Semaine du mois	W	Numérique	2
Jour de l'année	D	Numérique	189

Signification	Symbole	Présentation	Exemple
Jour du mois	d	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	a	Texte	pm
Heure de la journée (1 à 24)	k	Numérique	24
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	h	Numérique	12
Heure de la journée (0 à 23)	H	Numérique	0
Minutes dans une heure	m	Numérique	30
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978
Fuseau horaire	z	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Groupe de paramètres régionaux B

Paramètres régionaux : be-by, bg-bg, el, el-gr, fi, fi-fi, hr, hr-hr, hu, hu-hu, ja, ja-jp, ko, ko-kr, ro, ro-ro, ru, ru-ua, ru-ru, sh-yu, sk, sk-sk, sl-si, sq-al, sr-sp, th, tr, tr-tr, uk-ua, zh, zh-cn, zh-hk, zh-mo, zh-sg, zh-tw

Signification	Symbole	Présentation	Exemple
Ere	G	Texte	AD
Année	a	Numérique	2013
Année (de 'Semaine de l'année')	A	Numérique	2013
Mois de l'année	n	Texte et chiffres	Juillet et 07
Semaine de l'année	w	Numérique	27
Semaine du mois	W	Numérique	2
Jour de l'année	D	Numérique	189
Jour du mois	j	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	x	Texte	pm
Heure de la journée (1 à 24)	h	Numérique	24
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	k	Numérique	12
Heure de la journée (0 à 23)	H	Numérique	0
Minutes dans une heure	m	Numérique	30
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978

Signification	Symbole	Présentation	Exemple
Fuseau horaire	z	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Groupe de paramètres régionaux C

Paramètres régionaux : ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk

Signification	Symbole	Présentation	Exemple
Ere	G	Texte	AD
Année	u	Numérique	2013
Année (de 'Semaine de l'année')	U	Numérique	2013
Mois de l'année	M	Texte et chiffres	Juillet et 07
Semaine de l'année	w	Numérique	27
Semaine du mois	W	Numérique	2
Jour de l'année	D	Numérique	189
Jour du mois	t	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	a	Texte	pm
Heure de la journée (1 à 24)	h	Numérique	24

Signification	Symbole	Présentation	Exemple
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	k	Numérique	12
Heure de la journée (0 à 23)	H	Numérique	0
Minutes dans une heure	m	Numérique	30
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978
Fuseau horaire	z	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Groupe de paramètres régionaux D

Paramètres régionaux : de, de-at, de-be, de-ch, de-de, de-lu

Signification	Symbole	Présentation	Exemple
Ere	G	Texte	AD
Année	j	Numérique	2013
Année (de 'Semaine de l'année')	J	Numérique	2013
Mois de l'année	M	Texte et chiffres	Juillet et 07
Semaine de l'année	w	Numérique	27
Semaine du mois	W	Numérique	2
Jour de l'année	D	Numérique	189



Signification	Symbole	Présentation	Exemple
Jour du mois	t	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	a	Texte	pm
Heure de la journée (1 à 24)	h	Numérique	24
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	k	Numérique	12
Heure de la journée (0 à 23)	H	Numérique	0
Minutes dans une heure	m	Numérique	30
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978
Fuseau horaire	z	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Groupe de paramètres régionaux E

Paramètres régionaux : fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu

Signification	Symbole	Présentation	Exemple
Ere	G	Texte	AD
Année	a	Numérique	2013
Année (de 'Semaine de l'année')	A	Numérique	2013
Mois de l'année	M	Texte et chiffres	Juillet et 07
Semaine de l'année	w	Numérique	27
Semaine du mois	W	Numérique	2
Jour de l'année	D	Numérique	189
Jour du mois	j	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	x	Texte	pm
Heure de la journée (1 à 24)	h	Numérique	24
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	k	Numérique	12
Heure de la journée (0 à 23)	H	Numérique	0
Minutes dans une heure	m	Numérique	30

Signification	Symbole	Présentation	Exemple
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978
Fuseau horaire	z	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Groupe de paramètres régionaux F

Paramètres régionaux : ga-ie

Signification	Symbole	Présentation	Exemple
Ere	R	Texte	AD
Année	b	Numérique	2013
Année (de 'Semaine de l'année')	B	Numérique	2013
Mois de l'année	M	Texte et chiffres	Juillet et 07
Semaine de l'année	t	Numérique	27
Semaine du mois	T	Numérique	2
Jour de l'année	l	Numérique	189
Jour du mois	L	Numérique	10
Jour de la semaine dans le mois	F	Numérique	2 (deuxième mercredi de juillet)
Jour de la semaine (1=premier jour)	e	Numérique	2
Jour de la semaine	E	Texte	Mardi
Indicateur de matin ou d'après-midi (a.m. ou p.m.)	a	Texte	pm

Signification	Symbole	Présentation	Exemple
Heure de la journée (1 à 24)	u	Numérique	24
Heure du matin (a.m.) ou de l'après-midi (p.m.) (0 à 11)	K	Numérique	0
Heure du matin (a.m.) ou de l'après-midi (p.m.) (1 à 12)	k	Numérique	12
Heure de la journée (0 à 23)	U	Numérique	0
Minutes dans une heure	n	Numérique	30
Secondes dans une minute	s	Numérique	55
Milliseconde	S	Numérique	978
Fuseau horaire	c	Texte	Heure standard du Pacifique
Touche d'échappement utilisée dans le texte	'	n.d.	n.d.
Guillemet simple	"	n.d.	'

## Symboles des décimales

Tous les paramètres régionaux

Symbole	Signification
0	Chiffre qui est affiché même si la valeur est zéro.
#	Chiffre qui est supprimé si la valeur est zéro.
.	Paramètre fictif pour le séparateur décimal.
,	Paramètre fictif pour le séparateur des milliers.
E	Sépare la mantisse et l'exposant dans le format exponentiel.
;	Sépare les formats des nombres positifs et des nombres négatifs.
-	Préfixe négatif par défaut.

<b>Symbole</b>	<b>Signification</b>
%	Multiplié par 100, sous forme de pourcentage.
‰	Multiplié par 1 000, pour mille.
¤	Symbole de la devise. Si ce symbole est présent dans un motif, le séparateur décimal de la devise est utilisé à la place du séparateur décimal du système.
¤¤	Symbole de devise international. Il sera remplacé par un symbole de devise international. Si ce symbole est présent dans un motif, le séparateur décimal de la devise est utilisé à la place du séparateur décimal du système.
X	Autres caractères qui peuvent être utilisés dans le préfixe ou dans le suffixe.
'	Utilisé pour présenter des caractères spéciaux dans un préfixe ou dans un suffixe.
/u221E	Symbole d'infinité.
/uFFFD	Il ne s'agit pas d'un symbole numérique.



---

## Annexe F. Guide de gestion des données SAP BW à des fins d'utilisation dans Transformer

A partir de Transformer version 8.4, vous pouvez utiliser les packs Framework Manager publiés dans Content Manager pour tirer parti de vos données SAP BW. Les packs SAP peuvent être utilisés en tant que sources de données pour créer des modèles Transformer. Ainsi, les PowerCubes Transformer peuvent servir de méthodes de stockage en cache à accès rapide pour la diffusion de domaines plus restreints ou plus ciblés de vos informations métier. Cette méthode est recommandée pour tirer parti de vos données SAP BW.

Il existe des considérations spéciales à prendre en compte lors de l'utilisation des packs SAP créés dans Framework Manager. Pour en savoir davantage sur la création de requêtes SAP, sur la création de packs SAP dans Framework Manager et sur leur utilisation dans Transformer, reportez-vous à la section relative à l'utilisation de données BW SAP à l'aide d'un package dans Framework Manager. Pour obtenir des informations générales sur la création de packs dans Framework Manager, voir la section relative à la création ou à la modification d'un pack dans le *guide d'utilisation d'IBM Cognos Framework Manager*.

### Transformer versions 7.x, 8.1, 8.2 et 8.3

Dans Transformer versions 7.x, 8.1 et 8.2, vous pouvez tirer parti de vos données SAP BW via un pack Framework Manager dans lequel les sujets de requête et les dimensions sont externalisés via des fichiers CSV. Transformer peut utiliser les fichiers CSV en tant que source de données pour créer un modèle et générer des PowerCubes. Cette méthode ne doit être utilisée que dans un environnement IBM Cognos lorsque vous voulez exploiter les données d'IBM Cognos BI pour créer des PowerCubes pour IBM Cognos BI.

Dans Transformer version 8.3, utilisez les packs publiés pour les dimensions et les fichiers CSV pour les faits.

Il existe des considérations spéciales à prendre en compte lors de l'utilisation de fichiers CSV externalisés avec des données SAP dans Framework Manager. Pour en savoir davantage, reportez-vous à la section «Utilisation de données SAP BW à l'aide de fichiers CSV externalisés dans Framework Manager», à la page 661. Des informations générales sur l'externalisation des sujets de requête et des dimensions via la méthode CSV, consultez la section sur l'externalisation des sujets de requête et des dimensions dans le *guide d'utilisation de Framework Manager*.

---

## Utilisation de données SAP BW via un pack dans Framework Manager

Dans Transformer, vous pouvez tirer parti des données SAP BW en utilisant un pack SAP créé dans Framework Manager et publié dans Content Manager. Cette méthode est recommandée pour tirer parti de vos données SAP BW. Il existe des considérations spéciales à prendre en compte lors de l'utilisation des packs SAP créés dans Framework Manager.

Vous pouvez utiliser Transformer pour importer des données dimensionnelles et de faits d'une source de requête SAP BW. Les instructions ci-après décrivent la

procédure pour recréer un cube SAP BW en tant que cube IBM Cognos Transformer. Pour ce faire, le pack de requête SAP BW doit avoir un format spécifique.

L'importation d'une requête SAP BW pour accéder aux dimensions et aux faits via IBM Cognos BI s'effectue en trois phases :

- Création d'une requête dans SAP BW Business Explorer Query Designer
- Création d'un pack dans Framework Manager
- Création d'un modèle dans Transformer

### Restrictions

- Ce processus d'extraction est limité aux sources de données SAP BW uniquement.
- La source de données doit être une requête spécifiquement construite et définie dans la source de données SAP BW.

Des informations générales sur la création de packs figurent dans la section consacrée à la création et à la modification d'un pack dans le *guide d'utilisation de Framework Manager*.

## Création d'une requête BW dans SAP Business Explorer Query Designer

Vous devez créer une requête incluant le cube à importer. Il est recommandé de baser la requête sur un seul InfoCube dans la base de données. Une requête basée sur plusieurs sources peut aboutir à des erreurs SAP BW durant l'extraction des données.

Après avoir créé la requête, vous pouvez créer une variable «Création d'une variable», à la page 655.

### Procédure

1. Dans **Query Designer**, cliquez sur **New Query**.
2. Dans la boîte de dialogue **New Query**, sélectionnez le fournisseur d'informations contenant le cube à importer.
3. Cliquez sur l'icône **Outils** pour afficher le nom technique de l'**InfoObject**.
4. Faites glisser une caractéristique à importer depuis le catalogue **InfoObject** de la colonne de gauche vers l'une des zones sur la droite de la page. Par exemple, **Columns** ou **Rows**.

Les caractéristiques sélectionnées vont définir les métadonnées dans le cube Cognos Transformer. Elles doivent être conformes aux restrictions suivantes :

- Vous devez disposer d'au moins une variable facultative pour segmenter les données.
- Sélectionnez une caractéristique représentative de la source de données. Les caractéristiques peuvent être des chiffres clés, qui deviendront des mesures dans Cognos Transformer, ou des dimensions, qui deviendront les dimensions Cognos Transformer.
- N'affectez pas une hiérarchie d'affichage aux caractéristiques, que ce soit de façon explicite ou par le biais d'une variable.
- Tous les chiffres clés dans la requête SAP BW doivent être numériques.
- Ne sélectionnez pas la caractéristique **Currency/Unit**.
- Vérifiez que tous les chiffres clés sont dans la même devise.



- N'incluez dans la requête SAP BW que les caractéristiques à extraire via Framework Manager. L'inclusion de caractéristiques inutiles augmente le volume de données, ce qui a un impact négatif sur les performances.
  - Les caractéristiques doivent être copiées dans les zones **Colonnes** ou **Lignes** de la définition de requête. Si elles sont copiées dans les zones **Free Characteristics** ou **Filter**, les caractéristiques se présentent en tant que dimensions lors d'importation depuis le pack, mais le traitement d'extraction de flux ne peut pas extraire les valeurs.
  - Si des filtres sont définis, ils doivent référencer uniquement les dimensions qui ont été incluses à un autre endroit de la définition de requête.
  - Si vous incluez une caractéristique libre, aucune valeur ne va apparaître pour celle-ci dans l'extrait de chiffres clé. Un filtre sur une caractéristique libre agit comme un filtre sur les données SAP BW renvoyées. Vous pouvez l'utiliser comme filtre pour définir un sous-ensemble d'un InfoCube.
  - Utilisez une invite de sélection dans une liste de valeurs plutôt qu'une invite de saisie pour la requête. Une invite de sélection dans une liste de valeurs fournit des valeurs de segmentation de données.
5. Pour définir les métadonnées qui alimenteront le cube Cognos Transformer, vous devez modifier les propriétés de chaque caractéristique sélectionnée pour inclusion. Cliquez avec le bouton droit sur une caractéristique, puis sélectionnez **Properties**.
  6. Dans la boîte de dialogue **Properties of Characteristic**, définissez la valeur **Display As** sur **Key** et la valeur **Suppress Results Rows** sur **Always**. Notez que toute restriction ou filtre appliqué ici sera répercuté dans Cognos Transformer.
  7. Répétez les étapes 5 et 6 pour chaque caractéristique sélectionnée à l'étape 4.

**Remarque :** Ne sélectionnez que les caractéristiques requises. Pour éviter une consommation de mémoire excessive, ainsi que toute altération des performances du système ou toute panne de ce dernier, déterminez avec précision les caractéristiques à inclure dans la requête. Il est recommandé de consulter un administrateur SAP BW afin de vous assurer que les volumes de données ne sont pas dépassés.

8. Cliquez sur l'icône **Queries Properties** et, sous l'onglet **Extended**, cochez la case **Allow External Access to this Query**. Ainsi, la requête est accessible pour Framework Manager.
9. Cliquez sur **Save** et indiquez une **description** et un **nom technique** pour la nouvelle requête. Il est recommandé d'utiliser la convention de dénomination SAP BW dans la zone **Technical Name**. Débutez donc par la lettre "Z" suivie d'un nom intuitif ou conformément à votre convention de dénomination standard. Il est important de noter ce nom technique, car vous en aurez besoin pour trouver la requête dans Framework Manager.

## Résultats

Vous êtes prêt à créer une variable («Création d'une variable»). Pour en savoir davantage sur l'utilisation de **SAP Query Designer**, reportez-vous à la documentation SAP BW.

### Création d'une variable

Vous devez maintenant créer un paramètre d'invite facultatif pour la requête afin que Cognos Transformer puisse émettre des requêtes plus concises vers SAP en vue d'extraire l'intégralité de l'ensemble de données.

Il n'existe pas de règles définies pour l'utilisation des variables lors de l'extraction de données SAP BW à des fins d'utilisation dans Cognos Transformer. Toutefois, vous devez prendre garde à ne pas demander trop de données pouvant aboutir à de faibles performances ou à des erreurs avec des messages signalant une mémoire saturée dans votre environnement SAP.

L'approche de base de base à suivre est que, lorsqu'une variable est utilisée pour l'extraction, Cognos Transformer extraie d'abord tous les membres qui existent pour la dimension vis-à-vis de laquelle la variable est définie. Après quoi, Cognos Transformer devrait effectuer des extractions de données individuelles afin d'extraire les données de faits pour chacun des membres dans la dimension pour satisfaire la variable.

Cognos Transformer peut ainsi décomposer l'extraction de données en blocs gérables pouvant être traités par le serveur SAP BW. Il n'existe pas de norme définie concernant la dimension à laquelle l'appliquer. Pour obtenir des performances optimales, vous devez connaître vos données SAP BW et déterminer quelle dimension décompose de façon égale les données factuelles.

Vous devez choisir avec attention la dimension sur laquelle définir la variable. Il se peut que ces performances optimales ne soient obtenues qu'après quelques expérimentations. Par exemple, prenons une dimension Pays ou Région contenant trois pays ou régions en tant que membres, Etats-Unis (US), Canada (CA) et Mexique (MX). Si la plupart des affaires sont réalisées aux Etats-Unis (90 %), les transactions restantes (10 %) étant réparties de façon égale entre le Canada et le Mexique, cette dimension ne diviserait pas ces données en parts égales. Les requêtes résultantes comporteraient une demande très importante (US) et deux petites (CA et MX). Par conséquent, cette dimension ne serait pas intéressante.

Vous ne voulez pas appliquer une variable sur une dimension qui générerait trop de très petites demandes. Par exemple, [0MATERIAL], une dimension souvent utilisée dans les environnements SAP BW ne serait probablement pas intéressante car elle entraînerait la réalisation d'un trop grand nombre de petites demandes.

Vous pouvez disposer d'une dimension définie pour [CENTRECOUT] qui divise également les données pour 10 centres de coût distincts pouvant permettre de segmenter les données en part égales. Une autre possibilité admissible peut être une année ou un mois car vos données peuvent alors être divisées en sections ayant des performances adéquates.

Il n'est pas nécessaire d'appliquer des variables aux requêtes pour l'extraction de données. Certaines extractions peuvent s'effectuer parfaitement bien quand aucune variable n'est appliquée. Par exemple, une bonne approche peut consister à appliquer une variable sur une dimension qui divise les données en 20 extractions individuelles et à tester l'extraction. Si cela fonctionne bien, vous pouvez choisir d'appliquer une variable sur une dimension différente pouvant contenir 5 membres distincts afin de voir la différence.

Aucune formule ne peut être appliquée car tous les environnements sont différents. Toutefois, il est recommandé de procéder avec prudence afin d'éviter de perturber votre environnement SAP BW.

## Procédure

1. Dans **Query Designer**, cliquez avec le bouton droit de la souris sur une caractéristique sélectionnée dans la procédure précédente, puis sélectionnez **Restrict**.

Afin de vous assurer que les données sont réparties de façon égale, sélectionnez une caractéristique qui soit représentative du cube et ne résultera pas en un grand nombre de valeurs. La variable résultante voulue doit comporter un nombre de lignes pour chacune de ses valeurs qui soit similaire. Elle ne doit pas être à granularité trop fine (par exemple, peu de lignes par valeur résultant en un nombre excessif de requêtes), ni à granularité trop grossière (par exemple, plus d'un million de lignes par valeur).

2. Dans la boîte de dialogue **Selection for...**, cliquez sur l'onglet **Variables**, cliquez avec le bouton droit de la souris dans la fenêtre **Description** et sélectionnez **New Variable**.

**Remarque :** Si l'une des caractéristiques que vous avez choisies a déjà une variable, vous pouvez omettre l'étape de création d'une variable et passer à l'étape 7 de cette procédure.

3. Dans la page **New Variable Wizard General Information**, saisissez un **nom de variable** et une **description**, puis sélectionnez une dimension en tant que caractéristique. Cliquez sur **Next**.
4. Dans la page **Details**, sélectionnez **Single Value**, **Multiple Single Values** ou **Interval** dans la zone **Variable Represents**, sélectionnez **Optional** dans la zone **Variable entry is**, puis cochez la case **Ready for Input**. Cliquez sur **Suivant**.
5. Dans la page **Default Values**, vérifiez que la zone **Default Value** est vide.
6. Cliquez sur **Next** jusqu'à ce que vous soyez revenu à la boîte de dialogue **Selection for...** La nouvelle variable apparaît dans la fenêtre **Description**.
7. Sélectionnez la variable puis cliquez sur la flèche droite afin de placer la variable sélectionnée dans la fenêtre **Sélection**. Enregistrez la requête. Vous êtes prêt à importer la requête dans Framework Manager.

## Création d'un pack dans Framework Manager

Pour créer un pack dans IBM Cognos Framework Manager, vous devez :

- Importer les métadonnées SAP BW via l'assistant de métadonnées («Procédure d'importation à l'aide de l'assistant de métadonnées», à la page 658)

Framework Manager importe la requête SAP BW dans un modèle, et définit un pack qu'il exporte vers Content Manager.

Lors de l'importation, tenez compte des points suivants :

- Les dimensions sélectionnées dans la requête SAP BW sont disponibles dans les **dossiers de dimension** de la boîte de dialogue **Importer**.
  - Chaque dimension contient au moins une hiérarchie.
  - Sélectionnez toujours la hiérarchie principale dont le nom identifie la hiérarchie.
  - En présence d'autres hiérarchies, sélectionnez celle qui donne l'ensemble de niveaux souhaités au sein de la hiérarchie.
  - Framework Manager importe des dimensions de temps dans le modèle depuis la source de données SAP BW uniquement si un paramètre de configuration est activé. La définition de la configuration en tant que dimension de temps constitue une entrée globale ; chaque dimension importée sera alors traitée comme chaîne de date.
- Créer un pack («Procédure de création d'un pack», à la page 658)

Lors de la création du pack pour publication dans Content Manager, masquez la hiérarchie principale dans les dimensions où vous avez importé deux hiérarchies. La hiérarchie principale est requise et doit se trouver dans le pack

pour que l'utilisation de requêtes fonctionne correctement. Vous pouvez masquer la hiérarchie si vous ne voulez pas qu'elle soit visible.

## Procédure d'importation à l'aide de l'assistant de métadonnées

### Procédure

1. Dans Framework Manager, cliquez sur **Créer un nouveau projet**.
2. Complétez les zones de la boîte de dialogue **Nouveau projet**. Cliquez sur **OK**.
3. Suivez les étapes de l'**assistant de métadonnées**. Lorsque vous êtes invité à sélectionner une source de données, si vous avez besoin d'en créer une, cliquez sur **Nouveau....**
4. Dans la page **Sélectionner les objets**, recherchez la requête définie dans SAP BW Query Designer dans le cadre de la phase précédente («Création d'une requête BW dans SAP Business Explorer Query Designer», à la page 654). Faites défiler la liste jusqu'au nom technique fourni lorsque vous avez créé la variable. La structure du dossier est la suivante : Hiérarchies > Définition des niveaux > Définitions des éléments de requête.
5. Sélectionnez les éléments de requête principaux directement relatifs au niveau. C'est-à-dire ceux libellés **(Key)**, **(Name)**, etc.  
**Astuce** : Les attributs secondaires ou supplémentaires sont supprimés lors de l'importation dans Transformer. Seuls les éléments requis sont importés. Cependant, pour améliorer les performances, il est recommandé de ne pas sélectionner d'attributs secondaires ou supplémentaires. Si vous sélectionnez tous les attributs à ce stade, vous pouvez exclure les éléments de requête non souhaités lors de la publication du pack.
6. Complétez les écrans restants de l'**assistant de métadonnées**, en acceptant les valeurs par défaut, puis cliquez sur **Suivant**. Vous générez ainsi les dimensions et importez les métadonnées.
7. Sur le dernier écran de l'assistant, vérifiez les résultats, puis cliquez sur **Terminer**.

### Résultats

Vous êtes prêt à créer un modèle dans Transformer. Pour plus d'informations sur la création de packs, voir "Créer ou modifier un pack" dans le *guide d'utilisation* de Framework Manager.

## Procédure de création d'un pack

### Procédure

1. Cliquez sur le dossier **Packs** et, depuis le menu **Actions**, cliquez sur **Créer, Pack**.
2. Dans la page **Indiquer un nom**, saisissez le nom du pack et éventuellement une description et une infobulle. Cliquez sur **Suivant**.
3. Sélectionnez la requête importée dans la section précédente.  
Pour plus d'informations, voir «Création d'une requête BW dans SAP Business Explorer Query Designer», à la page 654.
4. Dans la page **Définir des objets**, lorsque vous masquez ou de excluez des objets enfant du pack, vous devez les sélectionner individuellement. L'exclusion d'objets parent entraîne celle de tous ses enfants. Notez que l'opération d'exclusion (ou de désélection) d'un grand nombre d'objets au niveau de cubes plus importants sera assez longue.

**Remarque :** Framework Manager prend en charge l'utilisation des combinaisons de touches [Ctrl]+Maj et [Alt]+Maj. Elles permettent de sélectionner plusieurs objets à inclure ou masquer dans le cube. Par exemple, si vous voulez inclure uniquement deux articles dans une succursale importante, sélectionnez-la totalité de celle-ci, puis utilisez la combinaison de touches [Ctrl]+Maj pour désélectionner les éléments à inclure et masquer les éléments sélectionnés restants.

Pour plus d'informations sur l'inclusion, l'exclusion et le masquage des objets, voir "Créer ou modifier un pack" dans le *guide d'utilisation* de Framework Manager.

5. Indiquez s'il faut utiliser les droits d'accès par défaut pour le pack :
  - Pour accepter les droits d'accès par défaut, cliquez sur **Terminer**.
  - Pour définir les droits d'accès, cliquez sur **Suivant**.
6. Lorsque vous êtes invité à ouvrir l'**assistant de publication de pack**, cliquez sur **Oui**.
7. Sélectionnez les valeurs par défaut et cliquez sur **Publier**. Le pack sera publié dans le magasin de contenu et vous pourrez y accéder dans Transformer.
8. Dans le dernier écran, vérifiez les résultats, puis cliquez sur **Terminer**.

## Résultats

Vous êtes prêt à créer un modèle dans Transformer. Pour plus d'informations sur la création de packs, voir "Créer ou modifier un pack" dans le *guide d'utilisation* de Framework Manager.

## Création d'un modèle dans Transformer

Transformer permet d'accéder à un pack SAP publié et à l'utiliser en tant que source de données pour créer un modèle. Une fois le modèle créé, vous pouvez créer des PowerCubes à utiliser avec le composant IBM Cognos souhaité, en accédant aux données dimensionnelles et aux données de faits depuis la source SAP BW d'origine. De plus, vous pouvez combiner les métadonnées SAP dans un modèle Transformer avec d'autres métadonnées d'entreprise ou des sources personnelles pourvu que vous ayez les informations clés correspondantes requises pour joindre les données durant la construction du cube.

Lorsque vous créez le modèle Transformer, vous devez utiliser l'Assistant **Insertion d'une dimension à partir d'un pack** au lieu de l'**Assistant Nouveau modèle**. Vous utilisez l'Assistant **Insertion d'une dimension à partir d'un pack** car il :

- crée une requête unique pour chaque dimension et pour les faits ;
- importe des faits et des dimensions de la même façon que les modèles relationnels modélisés de façon dimensionnelle (autrement dit, les faits et les dimensions sont importés en même temps) ;
- garantit que la portée est correctement définie entre les dimensions et les faits ;
- alimente la dimension avec la clé métier et les informations de légende appropriées ;
- importe uniquement les éléments nécessaires du pack BW requis pour la création du cube, lorsque les métadonnées sont importées. Cela permet de réduire le nombre d'attributs et de limiter les volumes de données aux seuls éléments requis pour la création du cube.

Si vous souhaitez définir des règles métier, utilisez le modèle Transformer plutôt que Framework Manager. Les calculs que vous définissez dans Framework Manager ne sont pas importés dans Transformer.

## Procédure de création d'un modèle dans Transformer

### Procédure

1. Dans Transformer, cliquez sur **Créer un modèle**.
2. Dans l'**assistant Nouveau modèle**, cliquez sur **Annuler**.
3. La sous-fenêtre **Grille des dimensions** étant sélectionnée, depuis le menu **Edition**, cliquez sur **Insérer une dimension à partir d'un pack**.
4. Cliquez sur **Parcourir** pour ouvrir le navigateur de métadonnées.
5. Dans la boîte de dialogue **Parcourir les métadonnées**, sélectionnez le pack qui contient votre requête SAP BW et cliquez sur **OK**.
6. Dans la boîte de dialogue **Insertion d'une dimension à partir d'un pack**, cliquez sur **Terminer**.
7. Dans la boîte de dialogue **Sélectionner les dimensions et les mesures à partir du pack**, cliquez sur les dimensions et les mesures à inclure dans la source de données.

Sélectionnez un élément de requête qui fournira les dates du PowerCube. Notez que les dates du PowerCube peuvent être dérivées entièrement des données de transaction.

8. Vous êtes notifié en cas d'erreurs ou d'avertissements. Dans la sous-fenêtre **Sources de données**, développez le pack pour afficher les requêtes et les éléments de requête de source de données. Les chiffres clés ou les mesures figurent dans la sous-fenêtre **Mesures**.

Assurez-vous que la règle d'agrégation de chaque mesure est correctement définie dans Transformer pour l'aligner aussi précisément que possible avec la règle d'agrégation définie dans SAP BW.

Il est recommandé de définir le type de stockage de toutes les mesures sur une virgule flottante de 64 bits.

Pour le niveau racine de chaque caractéristique (dimension), assurez-vous qu'il est marqué comme unique.

Les hiérarchies de présentation SAP BW peuvent contenir des chemins décalés, généralement associés aux noeuds " Non affecté " et " # " dans la hiérarchie. Les trous de ces hiérarchies produisent des vides au niveau associé dans la hiérarchie Transformer. Dans Transformer, il est possible de définir le texte qui doit être utilisé pour les espaces vides (le texte par défaut est "<espace vide>"). Il est conseillé de définir un texte plus approprié pour les entrées vides pour tous les niveaux de ce type.

9. Répétez les étapes 3 à 7 si vous voulez ajouter une autre requête.

**Astuce :** Si vous ajoutez une nouvelle dimension à un modèle après avoir ajouté les mesures, Transformer n'ajoute pas automatiquement la clé de la nouvelle dimension dans les chiffres clés. En conséquence, la portée n'est pas définie dans la grille des dimensions. Dans pareil cas, vous devez ajouter manuellement la clé de la nouvelle dimension aux chiffres clés en partant du niveau le plus bas de la dimension.

10. Enregistrez le modèle.

## Résultats

Vous pouvez à présent utiliser ce modèle pour créer des PowerCubes à utiliser avec le composant IBM Cognos souhaité, en accédant aux données dimensionnelles et aux données de faits depuis la source de données SAP BW d'origine. Pour en savoir davantage, reportez-vous à la section relative à la création d'un modèle du *guide d'utilisation* de Transformer.

---

## Utilisation de données SAP BW à l'aide de fichiers CSV externalisés dans Framework Manager

Lorsque vous externalisez des sujets de requête et des dimensions dans des formats utilisables dans d'autres applications, vous devez prendre en compte certaines considérations. Lors de l'extraction de données depuis SAP BW via Framework Manager, il est impératif de comprendre la distinction faite par Framework Manager entre les différents types de dimension. Chaque type de dimension présente un comportement différent lors de son externalisation et il peut être modifié avant cette opération.

Dans les versions 7.x, 8.1 et 8.2 de Transformer, vous pouvez optimiser vos données SAP BW en utilisant un pack Framework Manager dans lequel les sujets de requête et les dimensions sont externalisés à l'aide de fichiers CSV externalisés. Transformer peut utiliser les fichiers CSV en tant que source de données pour créer un modèle et générer des PowerCubes. Les fichiers CSV sont également pris en charge dans Transformer version 8.3, mais il est recommandé d'utiliser la prise en charge de pack pour les données dimensionnelles et les fichiers CSV pour les données de fait.

Dans la version actuelle de Transformer, l'utilisation des packs Framework Manager publiés dans Content Manager est la méthode recommandée pour optimiser les données SAP BW. Pour toute information générale sur la création de packs, voir "Create or Modify a Package" dans le document Framework Manager *User Guide*. Pour obtenir des informations spécifiques à SAP sur la création des packs, consultez la rubrique "Utilisation de données SAP BW via un pack dans Framework Manager".

### Taille d'extraction

La propriété de source de données **Taille d'extraction** dans Framework Manager contrôle la quantité de données extraites de SAP BW à tout moment.

Si ce paramètre est négatif, égal à zéro ou vide, une seule requête est émise vers SAP BW pour extraire les données de caractéristique.

Si la valeur de ce paramètre est positive, Framework Manager émet plusieurs requêtes vers SAP BW, chacune d'elles renvoyant approximativement le nombre de mégaoctets spécifié par la propriété **Taille d'extraction**.

Cette fonction peut réduire la taille globale du résultat de la requête sur le serveur SAP BW. L'exécution de la requête globale peut être plus longue, mais pour les caractéristiques importantes, le fait de ne pas utiliser cette fonction peut aboutir à la consommation de l'espace mémoire alloué à un utilisateur sur le serveur SAP BW.

L'intégralité des données pour une dimension de caractéristique sera en mémoire dans Framework Manager avant la production d'un fichier d'extraction. Il est important que seuls les éléments de requête requis soient extraits de SAP BW pour faire en sorte qu'une extraction n'échoue pas en raison d'erreurs d'allocation de mémoire dans Framework Manager.

Les sujets de requête de modèle sont extraits à l'aide du même mécanisme que celui utilisé pour l'exécution des requêtes dans IBM Cognos. Par conséquent, la propriété **Taille d'extraction** n'a aucune incidence sur l'exécution des requêtes.

## **Dimensions de mesure**

Lors de l'extraction d'une dimension de mesure, vous devez créer un sujet de requête de modèle contenant les mesures que vous souhaitez. Vous devez inclure l'élément de requête Clé métier de chacun des niveaux de chaque dimension, en fonction du niveau de granularité que vous cherchez à obtenir.

Pour plus d'informations sur l'externalisation des sujets de requête de modèle, voir «Considérations relatives à Framework Manager», à la page 664.

## **Dimensions de caractéristique**

Les dimensions de caractéristique sont externalisées indépendamment du type de la source de données SAP BW, tel qu'un InfoCube ou une requête SAP BW.

Framework Manager utilise une approche unique pour externaliser toutes les dimensions qui ne contiennent pas d'éléments de requête de fait. Dans de tels cas, le paramètre de configuration de taille d'extraction est utilisé pour contrôler la façon dont les données sont extraites de SAP BW.

**Remarque** : les sujets de requête de modèle sont externalisés d'une autre manière, sans tenir compte du fait qu'ils contiennent des éléments de requête de fait ou non. Pour plus d'informations sur l'externalisation des sujets de requête de modèle, voir «Considérations relatives à Framework Manager», à la page 664.

## **Dimensions de chiffres clés depuis un InfoCube SAP BW**

Lors de l'externalisation de la dimension de chiffres clés depuis un modèle basé sur un InfoCube, Framework Manager utilise exactement la même approche que celle utilisée pour l'externalisation des sujets de requête de modèle.

Si un InfoCube contenant plus de quelques milliers de transactions est externalisé directement à partir de Framework Manager, cela peut facilement provoquer un dépassement des limites de temps et de mémoire aussi bien sur le client et sur le serveur. Dans de tels cas, il est fortement recommandé d'utiliser une requête SAP BW comme base pour l'externalisation des métadonnées SAP BW.

## **Dimensions de chiffres clés depuis une requête SAP BW**

L'utilisation d'une requête BEx comme base pour l'externalisation de chiffres clés depuis une source de données SAP BW constitue, dans la plupart des cas, la meilleure approche. Si vous utilisez une variable BEx pour décomposer les données de la dimension de chiffres clés en sections gérables, des volumes importants de données de transaction définis arbitrairement peuvent être extraits de SAP BW.



Notez, cependant, que cette approche implique quelques restrictions concernant les données qui peuvent être extraites de SAP BW et la manière dont elles peuvent être extraites. La suite de cette section décrit la façon dont une requête SAP BW est utilisée pour extraire des données de SAP BW, ainsi que toutes les restrictions et limitations connues.

## Exigences relatives à une requête SAP BW

Pour le reste de cette section, on considère qu'une requête SAP BW est utilisée comme base pour l'externalisation des données et non comme base pour la génération de rapports ni dans l'intention de dépasser les limitations de mémoire et temporelles associées à l'extraction de données directement depuis un InfoCube.

Il n'est pas possible d'externaliser une requête SAP BW arbitraire. Une requête SAP BW doit être conforme aux restrictions ci-après si vous voulez l'externaliser :

- Définissez l'affichage des caractéristiques sur **Key** (Clé). Toute autre valeur peut générer des données incorrectes.

Pour modifier ce qui apparaît pour une caractéristique, cliquez sur celle-ci à l'aide du bouton droit de la souris, puis cliquez sur **Properties** (Propriétés). Dans la boîte de dialogue **Properties of Characteristic** (Propriétés des fonctions caractéristiques), définissez **Display As** (Afficher comme) sur **Key**.

Il est fortement recommandé d'utiliser **Key**.

- Pour réduire les volumes de données, ainsi que la quantité d'agrégation effectuée par le serveur SAP BW, il est fortement recommandé de désactiver la récapitulation pour toutes les caractéristiques de la requête dans sa feuille de propriétés.

Afin de désactiver la récapitulation pour une caractéristique, cliquez sur celle-ci avec le bouton droit de la souris le long de l'arête de la requête SAP BW et cliquez sur **Properties**. Dans la boîte de dialogue **Properties**, définissez **Suppress Results Rows** (Supprimer les lignes de résultat) sur **Always** (Toujours).

- Si au moins une caractéristique d'une requête SAP BW est affichée selon un autre mode que Key, la récapitulation pour toutes les caractéristiques doit être supprimée.
- La requête ne doit pas contenir la caractéristique **Currency/Unit** (Devise/Unité).
- Une hiérarchie d'affichage ne doit être affectée à aucune caractéristique, que ce soit explicitement ou via une variable.
- Si une caractéristique est incluse dans une requête SAP BW en tant que caractéristique libre, aucune valeur n'apparaîtra pour celle-ci dans l'extrait de données clés.

Un filtre sur une caractéristique libre agit comme un filtre sur les données renvoyées par SAP BW. Il s'agit d'un mécanisme efficace pour la définition d'un sous-ensemble d'un InfoCube.

Un tel filtre peut aussi être appliqué à une caractéristique le long d'un axe d'une requête SAP BW, auquel cas les valeurs filtrées apparaissent dans l'extrait de chiffres clés.

- Tous les chiffres clés de la requête SAP BW doivent être numériques.
- Les valeurs de chaque chiffre clé doivent être dans une seule devise. Une variable ne doit pas être utilisée pour effectuer l'affectation d'une devise cible.
- N'incluez dans la requête SAP BW que les caractéristiques à extraire via Framework Manager. L'inclusion de caractéristiques superflues augmente le volume de données transféré depuis SAP BW, affectant ainsi les performances.

## Instructions et contraintes liées à l'utilisation des cubes SAP BW

Vous devez utiliser des fichiers CSV lors de l'importation de métadonnées à partir de cubes SAP BW. Pour des raisons de performance, nous vous conseillons d'effectuer un filtrage au niveau géographique, temporel ou d'une autre dimension afin de limiter la quantité de données extraites. N'oubliez pas d'appliquer votre filtre de dimension aux dimensions associées et à leurs tables de faits (mesures).

Pour plus d'informations, voir la section "Create a Filter" dans le document *Framework Manager User Guide*.

Les cubes SAP BW étant multidimensionnels, des cumuls sont appliqués à la source. Si vous modifiez le type de cumul après l'importation des données dans Cognos Transformer, les résultats que vous obtiendrez ne seront pas valides.

Des données manquantes ou des métadonnées hors de portée pour une mesure donnée peuvent entraîner générer des résultats différents, en fonction du contexte. Vous pouvez obtenir les résultats suivants :

- Des valeurs nulles
- Des symboles #
- REST\_H
- La mention Non affecté

De tels jetons en double pouvant générer des problèmes dans Cognos Transformer, dans les niveaux uniques par exemple, il est recommandé d'affecter des filtres à la dimension afin qu'ils n'apparaissent pas dans les données importées.

Enfin, prenez soin de ne sélectionner que les éléments de requête requis pour générer vos données filtrées.

## Considérations relatives à Framework Manager

Lors de l'extraction de la dimension de mesure d'une requête SAP BW, la propriété **Taille d'extraction** de la source de données contrôle la quantité de données extraite du serveur SAP BW à un moment donné. Les sujets de requête de modèle sont externalisés d'une autre manière, sans tenir compte du fait qu'ils contiennent des éléments de requête de fait ou non. Dans ce scénario, ce paramètre n'a aucun effet sur le serveur SAP BW, mais il limite la quantité de mémoire allouée par Framework Manager à un moment donné pour extraire les données.

Notez que les filtres définis sur la dimension de chiffres clés ne sont pas appliqués lors de l'extraction de données d'une requête SAP BW. Pour bénéficier des performances liées à l'extraction de données d'une requête SAP BW, vous devez définir des filtres dans une requête SAP BW.

De plus, les éventuels calculs définis dans la dimension de chiffres clés sont ignorés. Ces calculs peuvent être définis au sein de la requête SAP BW dans BEx ou dans un sujet de requête de modèle dans Framework Manager.

Chaque caractéristique extraite doit contenir au moins un élément de requête provenant du niveau le plus bas de sa hiérarchie (s'il en existe un) afin de fournir un lien vers l'extrait de chiffres clés. Vous devez inclure l'élément de requête Clé métier de chacun des niveaux de chaque dimension, en fonction du niveau de granularité que vous cherchez à obtenir.

## Utilisation de variables pour externaliser des données clés depuis une requête SAP BW

Le volume de transactions dans une requête SAP BW est tel que, dans la plupart des cas, l'utilisation d'une seule requête pour extraire les données de SAP BW va consommer plus de mémoire que celle allouée à un utilisateur sur un serveur SAP BW. Dans Framework Manager, vous pouvez utiliser une variable facultative pour extraire les données de fait d'une requête SAP BW dans des sections de taille raisonnable.

Pour utiliser cette fonction, une caractéristique incluse dans la requête SAP BW (mais pas en tant que caractéristique libre) se voit affecter une variable conforme aux restrictions suivantes :

- Il doit s'agir d'une seule valeur.
- Elle doit être facultative.
- Elle ne doit pas comporter de valeur par défaut.
- Elle peut être définie sur la caractéristique ou une hiérarchie de présentation.

Si une requête SAP BW contient une telle variable et que la dimension de données clés est externalisée, Framework Manager exécute une requête pour chaque valeur possible associée à une variable. Par conséquent, en choisissant une caractéristique appropriée, la dimension de données clé peut être extraite sans dépasser les restrictions de mémoire du client ou du serveur. Les caches sur le client et le serveur sont réinitialisés après chaque requête.

Si une hiérarchie de présentation est utilisée pour effectuer la création de sections d'extraction, il est important que les valeurs pour une variable soient obtenues depuis un seul niveau dans la hiérarchie, sinon l'extrait contiendra des données récapitulées à différents niveaux. Afin de limiter les valeurs pour une variable à un seul niveau d'une hiérarchie, éditez la restriction de niveau de la variable dans Framework Manager. Par exemple, l'utilisation d'une valeur comme "2:2" indique que seules les valeurs du second niveau de la hiérarchie doivent être utilisées (le niveau 0 est la racine d'une hiérarchie).

En présence d'une requête SAP BW dotée d'une telle variable, la valeur de la variable est réinitialisée après chaque requête.

Si une requête SAP BW contient autre chose qu'une variable, ou une variable définie d'une autre façon que ci-dessus, Framework Manager ne tente pas d'utiliser une variable pour diviser l'extraction de la dimension des données clés en sections de plus petite taille.

### Solutions aux problèmes liés à l'externalisation

Lors de l'externalisation d'une source de données depuis Framework Manager, une erreur d'authentification peut se produire si

- le modèle est publié dans Content Manager,
- l'exécution de l'externalisation des données dépasse le délai affecté aux passeports dans IBM Cognos Configuration.

Les utilisateurs ne sont pas invités à saisir de nouveau leurs données d'identification.

Si une erreur se produit, les données externalisées restent complètes et valides. Cependant, si le modélisateur choisit de publier le modèle, il doit authentifier et publier de nouveau le modèle, mais sans externaliser les données.

Une autre solution consiste à publier le modèle sur le réseau, auquel cas l'erreur d'authentification ne se produit pas.

## Création de PowerCubes à partir de données SAP BW

Vous pouvez créer des PowerCubes IBM Cognos à partir de données SAP BW. Il existe des instructions à suivre pour Framework Manager et pour Transformer.

Pour plus d'informations, voir «Instructions relatives à Framework Manager» et «Instructions pour Transformer».

### Instructions relatives à Framework Manager

Lorsque vous externalisez des données dans le but de créer un ou plusieurs PowerCubes, tenez compte des considérations ci-dessous.

- L'extrait de chaque caractéristique doit avoir un élément de requête de clé commune qui soit équivalent à un élément de requête de clé secondaire dans l'extrait de chiffres clés.
- Pour un extrait basé sur une requête SAP BW, il est fortement recommandé que toutes les caractéristiques soient affichées en mode clé (Key) dans la requête SAP BW.
- Si une caractéristique n'a pas de hiérarchie de présentation ou si une nouvelle est souhaitée, procédez à l'extraction d'un ou de plusieurs éléments de requête pouvant constituer la base des niveaux dans une hiérarchie.
- Pendant l'importation de métadonnées SAP BW dans un modèle qui extraira des données, limitez le modèle aux seuls éléments de requête absolument indispensables pour créer un PowerCube. Les performances d'extraction des données s'en trouveront améliorées.
- Les valeurs nulles sont insérées dans les fichiers CSV lorsque vous externalisez des dimensions et des sujets de requête SAP BW.
- Une limite pratique pour les PowerCubes est 2 000 000 catégories (valeurs) pour une dimension (caractéristique).

### Instructions pour Transformer

Lors de l'utilisation de données SAP BW extraites de Framework Manager, gardez ces considérations à l'esprit.

- Dans Transformer version 8.3, vous pouvez insérer des dimensions ordinaires à partir de sources de données SAP provenant directement d'une source de données IBM Cognos, via l'option **Insérer une dimension à partir d'un pack**.
- Par le biais de l'assistant de modèle dans Transformer, insérez une source de données de type Texte à zones délimitées comportant des titres de colonne et commencez par sélectionner le fichier CSV. N'exécutez pas la conception automatique.
- Faites glisser toutes les colonnes de chiffres clés de la sous-fenêtre Sources de données dans la sous-fenêtre Mesures. Assurez-vous que la règle d'agrégation de chaque mesure est correctement définie dans Transformer pour l'aligner aussi précisément que possible avec la règle d'agrégation définie dans SAP BW.
- Il est recommandé de définir le type de stockage de toutes les mesures sur une virgule flottante de 64 bits.
- Par le biais de l'assistant Dates, sélectionnez un élément de requête qui fournira les dates du PowerCube. Notez que les dates du PowerCube peuvent être dérivées entièrement des données de transaction.
- Insérez les différents fichiers CSV correspondant aux caractéristiques qui ont été externalisées via Framework Manager.

Chaque fichier CSV contient une colonne qui correspond à une colonne dans le fichier CSV de données clés. En cliquant avec le bouton droit de la souris sur les diverses colonnes et en éditant leurs propriétés, vérifiez que les colonnes fournissant un lien entre une caractéristique et les chiffres clés ont le même nom. Par exemple, si une colonne de chiffres clés est intitulée Client et que la colonne correspondante dans le fichier CSV du client est intitulée Client - Clé, le nom de la colonne dans le fichier CSV de chiffres clés peut être modifié en Client - Clé.

- Pour chaque caractéristique, créez une dimension en utilisant les colonnes clés ou d'autres attributs d'une caractéristique, pour contrôler les niveaux de la dimension. Pour chaque niveau, vérifiez que des colonnes source sont affectées aux propriétés pour le libellé, le nom abrégé et la description (le cas échéant).
- Pour le niveau racine de chaque caractéristique (dimension), assurez-vous qu'il est marqué comme unique.
- Les hiérarchies de présentation SAP BW peuvent contenir des chemins décalés, généralement associés aux noeuds " Non affecté " et " # " dans la hiérarchie. Les trous de ces hiérarchies produisent des vides au niveau associé dans la hiérarchie Transformer.

Dans Transformer, il est possible de définir le texte qui doit être utilisé pour les espaces vides. Le texte par défaut est blank, mais vous devez définir un texte plus approprié pour les entrées vides pour tous les niveaux de ce type.



---

## Annexe G. Mots réservés

Vous devez veiller à ce que les noms des sources de données, des tables et des colonnes n'utilisent pas de noms réservés par le logiciel IBM Cognos.

Si vous devez utiliser un mot réservé, placez-le entre guillemets dans la spécification SQL. Par exemple, `select Orderdate, "Timezone"`.

Les mots suivants sont réservés :

Plage	Mots réservés
A à C	abs, all, and, any, as, asc, at, avg, between, bigint, bit_length, boolean, both, by, call, case, cast, ceil, ceiling, char, char_length, character, character_length, coalesce, count, create, cross, cube, cume_dist, current, current_date, current_time, current_timestamp, cursor
D à M	date, day, dbkey, dec, decimal, declare, delete, dense_rank, desc, distinct, double, else, end, escape, except, exists, exp, extract, false, filename, filter, first, first_value, float, floor, following, for, from, full, group, grouping, hash, having, hour, in, inner, insert, int, integer, intersect, interval, into, is, join, last, last_value, leading, left, like, ln, local, localtime, localtimestamp, loop, lower, max, merge, min, minute, mod, month
N à Q	national, natural, nchar, no, not, ntile, null, nullif, nulls, numeric, nvarchar, octet_length, of, on, or, order, others, out, outer, over, partition, perc, percent_rank, percentile, percentile_cont, percentile_disc, position, power, preceding, precision, qualify
R à T	range, rank, ratio_to_report, ravg, rcount, rdiff, real, recursive, returning, right, rmax, rmin, rollup, row, row_number, rows, rsum, scroll, second, select, set, sets, smallint, snapshot, some, sqrt, stddev, stddev_pop, stddev_samp, substring, sum, table, tertile, then, time, timestamp, timezone_hour, timezone_minute, to, top, trailing, trim, true,
U à Z	unbounded, union, unknown, updatable, update, upper, user, using, values, var_pop, var_samp, varchar, variance, varying, when, where, window, with, within, without, xavg, xcount, xfirst, xlast, xmax, xmin, xmovingavg, xmovingsum, xntile, xperc, xrank, xratio, xstddev, xstddev_pop, xsum, xtertile, xvariance, xvariance_pop, year, zone

Les mots suivants sont également des mots réservés : `_cursor`, `_local` et `_rowset`.





---

## Annexe H. Types de données XML

Vous pouvez importer des données XML sous la forme d'une source de données tabulaires dans IBM Cognos Framework Manager.

Les types de données suivants sont pris en charge lors de l'importation de données depuis XML vers Framework Manager.

- boolean
- byte
- date
- dateTime
- decimal
- double
- ENTITIES
- ENTITY
- float
- ID
- IDREF
- int
- integer
- language
- long
- Nom
- NCName
- negativeInteger
- NMTOKEN
- NMTOKENS
- nonNegativeInteger
- NonPositiveInteger
- NOTATION
- positiveInteger
- QName
- short
- string
- time
- token
- unsignedLong
- unsignedInt
- unsignedShort
- unsignedByte

Les types de données suivants ne sont pas pris en charge.

- base64Binary
- duration

- gYearMonth
- gYear
- gMonthDay
- gMonth
- hexBinary

---

## Remarques

Le présent document concerne des produits et des services disponibles dans différents pays.

Cet élément peut être mis à disposition par IBM dans d'autres langues. Vous pouvez toutefois devoir détenir une copie du produit ou une version du produit dans cette langue pour pouvoir y accéder.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM. Le présent document peut décrire des produits, des services ou des fonctions qui ne sont pas inclus dans le Logiciel ni dans l'autorisation d'utilisation que vous avez acquise.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations  
IBM Canada Ltd  
3600 Steeles Avenue East  
Markham, Ontario  
L3R 9Z7  
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit auprès d' IBM à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Software Group  
Attention: Licensing  
3755 Riverside Dr.  
Ottawa, ON K1V 1B7  
Canada

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer

l'exactitude de leurs performances ni leur compatibilité. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Selon la configuration déployée, la présente Offre Logiciels peut utiliser des cookies de session et des cookies persistants destinés à collecter

- le nom
- le nom d'utilisateur
- le mot de passe

à des fins

- de gestion de session
- d'authentification
- de facilité d'utilisation des produits
- de configuration d'un code d'accès unique
- de suivi de l'utilisation, ou pour des fonctions autres que celles-ci.

Ces cookies ne peuvent pas être désactivés.

Si les configurations déployées pour cette offre logicielle vous fournissent, en tant que client, la possibilité de collecter des informations identifiant l'utilisateur final via des cookies ou d'autres technologies, vous devez vérifier auprès de votre conseiller juridique les lois applicables en matière de collecte de données, y compris concernant vos obligations d'information préalable et de consentement.

Pour plus d'informations sur l'utilisation à ces fins de ces différentes technologies, dont les cookies, voir les règles de confidentialité IBM sur <http://www.ibm.com/privacy> et la politique sur la protection des renseignements personnels en ligne d'IBM sur <http://www.ibm.com/privacy/details> dans la section nommée "C: Cookies, balises Web et autres technologies" ainsi que "Déclaration de confidentialité des produits logiciels IBM et des logiciels en tant que service (SaaS)" sur <http://www.ibm.com/software/info/product-privacy>.

---

## Marques

IBM, le logo IBM et [ibm.com](http://www.ibm.com) sont des marques d'International Business Machines Corp., dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web «Copyright and trademark information » à [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Les termes qui suivent sont des marques d'autres sociétés :

- Adobe, le logo Adobe, PostScript et le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.
- Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.
- UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.
- Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.

---

## Glossaire

Ce glossaire contient les termes et les définitions d'IBM Cognos Business Intelligence et d'IBM Cognos Framework Manager.

Les références croisées suivantes sont utilisées dans ce glossaire :

- Voir fait référence, dans le cas d'un terme, à un synonyme préféré ou, dans le cas d'un acronyme ou d'une abréviation, à la forme complète définie.
- Voir aussi fait référence à un terme lié ou opposé.

Pour afficher les glossaires des autres produits IBM, accédez à [www.ibm.com/software/globalization/terminology](http://www.ibm.com/software/globalization/terminology) (s'ouvre dans une nouvelle fenêtre).

«A» «C», à la page 678 «D», à la page 679 «E», à la page 679 «F», à la page 680 «G», à la page 681 «H», à la page 681 «I», à la page 681 «L», à la page 681 «M», à la page 681 «N», à la page 682 «O», à la page 682 «P», à la page 682 «R», à la page 683 «S», à la page 684 «T», à la page 685 «U», à la page 685 «W», à la page 685 «Z», à la page 685

---

### A

#### accès anonyme

Type d'accès permettant aux utilisateurs et aux serveurs d'accéder à un serveur sans avoir besoin d'authentification auprès de celui-ci.

**agent** Processus qui effectue une action au nom d'un utilisateur ou d'un autre programme sans intervention de l'utilisateur ou de manière régulière et transmet les résultats à l'utilisateur ou au programme.

#### algorithme de cryptographie

Combinaison des algorithmes d'authentification, d'échange de clés et de la spécification de chiffrement du protocole SSL utilisée pour l'échange de données sécurisé.

**alias** Autre nom utilisé à la place du nom principal.

#### arborescence de données

Dans un studio, structure qui contient des objets tels que des sujets de requêtes, des éléments de requêtes, des dimensions, des niveaux et des membres. Une arborescence de données est utilisée en tant que palette des données disponibles qui peuvent être insérées dans des calculs, des filtres, des zones d'affichage et d'autres opérations de création de rapports.

#### archive de déploiement

Fichier utilisé pour le déploiement. Une archive de déploiement contient les données du magasin de contenu qui sont déplacées.

#### article de forum

Entrée unique dans un format compatible RSS (Really Simple Syndication). format. Elle peut inclure un titre, du texte et un lien pour obtenir plus d'informations. Une tâche d'article de forum dans un agent peut être utilisée pour créer des articles de forum à afficher dans un portlet Cognos Connection.

#### attribut

1. Dans le cas de modèles dimensionnels, propriété qui fournit des informations qualitatives à propos des membres d'un niveau dans une dimension. Par exemple, le niveau Magasin dans la dimension Détaillant peut avoir des propriétés telles que l'adresse ou la succursale. En général, aucune valeur de mesure ou aucun cumul n'est associé aux attributs dimensionnels; ainsi, ces derniers sont utilisés pour trouver des membres ou pour appliquer des filtres à ceux-ci.
2. Dans le cas de modèles relationnels, il s'agit d'une requête qui n'est pas une mesure ou un identificateur. Lorsque un élément de requête est un attribut, il ne devrait pas être agrégé ou utilisé aux fins de groupage ou de création de listes déroulantes d'invites.
3. Dans la modélisation BI, il s'agit d'une caractéristique descriptive d'une entité

plutôt que d'un identificateur unique ou d'une mesure récapitulative.

#### **authentification**

Processus de validation de l'identité d'un utilisateur ou d'un serveur.

#### **autorité de certification (CA)**

Composant qui émet des certificats sur chaque poste de travail où sont installés des composants.

---

## **C**

**CA** Voir autorité de certification.

#### **cardinalité**

1. Dans des sources de données relationnelles, indication numérique de la relation entre deux sujets de requête, éléments de requête ou autres objets de modèle.
2. Nombre de membres d'une hiérarchie dans des sources de données OLAP. La propriété de cardinalité d'une hiérarchie est utilisée pour assigner un ordre de résolution aux expressions.

#### **carte d'information**

Affichage des informations générales concernant le contenu d'un tableau de bord, d'un espace de travail ou d'un rapport, telles que le propriétaire, les informations de contact, la date de modification, ainsi qu'une miniature du tableau de bord ou du rapport.

#### **carte stratégique**

Dans Metric Studio, représentation visuelle de la stratégie et des objectifs de cette stratégie pour une organisation. Par exemple, une carte stratégique peut illustrer la place qu'occupe le travail des employés dans les objectifs globaux de l'organisation.

#### **certificat**

Pour la sécurité de l'ordinateur, document numérique qui associe une clé publique à l'identité du propriétaire du certificat, permettant ainsi au titulaire du certificat d'être authentifié. Un certificat est émis par une autorité de certification et est signé numériquement par cette dernière. Voir aussi autorité de certification.

**CGI** Voir interface CGI.

#### **clé de diffusion en rafale**

Dimension ou niveau d'une requête dans une spécification de rapport qui permet de créer ou de diffuser en rafale un ensemble de résultats de rapports.

#### **clé d'événement**

Combinaison de données élémentaires qui définit l'instance d'un événement de façon unique. L'identification de l'instance d'un événement permet à l'agent de déterminer s'il s'agit d'une instance nouvelle, en cours ou arrêtée.

**CM** Voir Content Manager.

#### **colonne définie par l'utilisateur**

Dans le contexte de la gestion d'indicateurs, colonne qui représente une valeur autre que la valeur réelle ou cible. Il peut s'agir d'une valeur de référence dans l'industrie ou de toute autre information numérique supplémentaire pour une période, incluant un calcul basé sur les autres valeurs d'un indicateur. Les colonnes définies par l'utilisateur peuvent être différentes pour chaque type d'indicateur.

#### **composant du groupe de serveurs d'application**

Lors de l'installation, l'ensemble des processeurs qui accèdent aux bases de données de requêtes pour recueillir des informations et afficher les résultats sous forme de rapports en format PDF ou HTML et sous forme d'indicateurs. Les composants du groupe de serveurs d'applications envoient aussi les demandes à Content Manager et affichent les résultats que Content Manager extrait du magasin de contenu.

#### **condition**

Expression pouvant être évaluée comme étant vraie, fausse ou inconnue. Elle peut être exprimée sous forme de texte en langage naturel, en notation formelle mathématiquement ou sous forme de langage lisible par machine.

#### **connexion de source de données**

Information nommée qui définit le type de source de données, son emplacement physique et toute exigence en termes de code d'accès. Une source de données peut disposer de plusieurs connexions.

#### **contact**

Adresse de courrier électronique nommée



à laquelle des rapports et des courriers électroniques d'agent peuvent être envoyés. Les contacts ne sont jamais authentifiés.

#### **Content Manager (CM)**

Service qui extrait des informations du magasin de contenu et y enregistre des informations.

#### **Content Store**

Base de données qui contient les données nécessaires au fonctionnement d'une application, telles que les spécifications de rapports, les modèles publiés et les droits de sécurité.

#### **contrainte**

1. Spécification de sécurité empêchant un ou plusieurs utilisateurs d'accéder au composant d'un modèle ou d'exécuter une tâche de modélisation ou de création.
2. Restriction relative aux valeurs que les utilisateurs peuvent saisir dans une zone.

**cube** Représentation multidimensionnelle des données nécessaire aux applications de traitement analytique en ligne, de reporting multidimensionnel ou de planification multidimensionnelle.

---

## **D**

#### **déploiement**

Processus consistant à déplacer une application (telle qu'un rapport ou un modèle) vers une instance différente. Par exemple, les rapports sont souvent créés dans un environnement de test, puis déployés dans un environnement de production. Lorsqu'une application est déployée, elle est exportée, transférée, puis importée.

#### **diffuser en rafale**

Produire plusieurs résultats de rapports en exécutant un rapport une seule fois. Par exemple, l'utilisateur peut créer un rapport qui présente les ventes effectuées par chaque employé et l'exécuter une seule fois, en effectuant du même coup une diffusion en rafale qui envoie à chaque responsable régional les résultats propres à sa région.

#### **dimension**

Groupement élargi de données descriptives concernant un aspect essentiel d'une entreprise, tel que des produits, des dates ou des emplacements. Chaque dimension inclut différents niveaux de membres dans une ou plusieurs hiérarchies, ainsi qu'un ensemble facultatif de membres calculés ou de catégories spéciales.

#### **dimension conforme**

Dimension, avec une seule définition, qui peut être réutilisée ou partagée par plusieurs magasins de données coordonnées.

#### **données d'identification**

Ensemble d'informations qui accorde à un utilisateur ou à un processus certains droits d'accès.

#### **droit d'accès**

Privilège permettant l'accès ou l'utilisation d'un objet.

---

## **E**

#### **élément**

Voir membre.

#### **élément de requête**

Représentation d'une colonne de données dans une source de données. Les éléments de requêtes apparaissent dans un modèle ou dans un rapport et contiennent une référence à une colonne de base de données, une référence à un autre élément de requête ou un calcul.

#### **ensemble**

Collection d'éléments ou de membres qui ont quelque chose en commun. Les membres de l'ensemble peuvent être sélectionnés de façon spécifique ou bien à l'aide d'une ou plusieurs règles de filtrage. Voir aussi ensemble personnalisé, ensemble basé sur des détails, ensemble de pages, ensemble prédéfini, ensemble basé sur une sélection, ensemble superposé.

#### **ensemble basé sur des détails**

Ensemble basé sur un élément et ses détails immédiats. Voir aussi ensemble.

#### **ensemble basé sur une sélection**

Collection d'éléments individuels que l'utilisateur a spécialement sélectionnés.

Les éléments ou les membres peuvent être sélectionnés à partir d'un ou de plusieurs niveaux d'une même hiérarchie. Voir aussi ensemble.

**ensemble de pages**

Dans Report Studio, ensemble d'une ou plusieurs pages conçues afin d'être reproduites dans la sortie de rapport pour chaque instance d'un élément de requête donné. Voir aussi ensemble.

**ensemble d'union**

Voir ensemble superposé.

**ensemble nommé**

Voir ensemble prédéfini.

**ensemble personnalisé**

Dans Analysis Studio, objet nommé pouvant contenir des règles de filtrage, des calculs et des règles de tri. Les ensembles personnalisés peuvent définir un ensemble de membres différent de ceux définis à l'origine dans le modèle de cube. Voir aussi ensemble prédéfini, ensemble.

**ensemble prédéfini**

Ensemble de membres défini à l'intérieur d'une source de données OLAP sous forme de liste ou par une expression. Les ensembles prédéfinis peuvent être utilisés dans les opérations d'analyse et lors de la création de rapports. Voir aussi ensemble personnalisé, ensemble.

**ensemble superposé**

Plusieurs ensembles superposés à l'horizontale ou à la verticale. Voir aussi ensemble.

**environnement local du produit**

Code ou paramètre qui indique la langue ou les paramètres régionaux à utiliser pour certaines sections de l'interface du produit, telles que les commandes de menu.

**espace de travail**

Voir tableau de bord.

**espace-noms**

Partie du modèle dans lequel les noms peuvent être définis et utilisés. Dans un espace-noms, chaque nom a une signification unique.

**étape de travail**

La plus petite partie d'un travail pouvant

être exécutée séparément. Une étape de travail peut être un rapport ou un autre travail.

**événement**

Changement d'état, tel que la fin ou l'échec d'une opération, d'un processus métier ou d'une tâche humaine qui peut déclencher une action telle que la sauvegarde des données de l'événement dans un référentiel de données ou l'appel d'un autre processus métier.

**extrait d'indicateurs**

Ensemble de mappages entre une source de données Cognos et un objet ou une valeur de Metric Studio. Par exemple, la mesure de cube Revenus est mappée avec l'indicateur Revenus - Valeurs réelles de Metric Studio.

**extrait d'objets**

Extrait qui définit les métadonnées d'un objet Metric Studio, telles qu'une colonne définie par l'utilisateur, un scorecard ou une source de données.

---

**F**

**fait** Voir mesure.

**fichier de réponses**

Fichier ASCII pouvant être personnalisé avec les données d'installation et de configuration qui automatisent l'installation. Les données d'installation et de configuration doivent généralement être entrées lors d'une installation interactive, tandis que le fichier de réponse permet d'effectuer l'installation sans la moindre intervention.

**fonction**

Groupe de fonctions et fonctionnalités pouvant être masquées ou affichées afin de simplifier l'interface utilisateur. Les fonctions peuvent être activées ou désactivées en changeant la configuration des paramètres préférentiels ou contrôlés par le biais d'une interface d'administration.

**fournisseur d'authentification**

Mécanisme de communication avec une source d'authentification externe. Les fonctions d'authentification des utilisateurs, d'inscription à des groupes et de recherche d'espaces-noms sont

disponibles par l'intermédiaire des fournisseurs d'authentification.

#### **fournisseur de sécurité**

Voir fournisseur d'authentification.

---

## **G**

### **glyphe**

La forme réelle (profil binaire, nomenclature) de l'image d'un caractère. Par exemple, A italique et A roman sont deux glyphes différents représentant le même caractère sous-jacent. En d'autres termes, deux images dont la forme est différente représentent deux glyphes distincts. Dans le cadre de l'utilisation qui va en être faite, le glyphe est un synonyme de l'image du caractère ou tout simplement de l'image (Norme Unicode - Version 1.0).

### **groupe**

Ensemble d'utilisateurs qui peuvent partager des droits d'accès à des ressources protégées.

### **groupement**

En matière de reporting, il s'agit du regroupement des valeurs communes des éléments de requête afin de ne les afficher qu'une seule fois.

---

## **H**

### **hiérarchie**

Organisation d'un ensemble d'entités en arborescence, chaque entité (à l'exception de la racine) ayant une ou plusieurs entités parents et un nombre arbitraire d'entités enfant.

---

## **I**

### **index dérivé**

Indicateur calculé qui fournit un statut et un score basés sur d'autres indicateurs.

### **indicateur**

Mesure pour évaluer la performance dans un domaine clé de votre société.

### **initiative**

Tâche développée pour atteindre des objectifs ou combler l'écart entre les performances et les cibles. Les initiatives

sont associées à des objectifs individuels et prennent souvent la forme de projets, actions ou activités.

### **interface CGI**

Norme Internet de définition des scripts qui transmettent les informations d'un serveur Web à un programme d'application par le biais d'une demande HTTP et inversement.

**invite** Élément de rapport qui demande des valeurs avant l'exécution du rapport.

### **invite en cascade**

Invite qui utilise les valeurs d'une invite précédente pour filtrer les valeurs de l'invite ou de la liste de valeurs en cours.

---

## **L**

### **langage d'expressions multidimensionnelles (MDX)**

MDX est l'équivalent multidimensionnel du langage SQL.

### **liste des événements**

Ensemble d'instances détectées d'un événement qui sont évaluées par les règles d'exécution des tâches pour déterminer quelle tâche d'agent devrait être exécutée.

### **liste de surveillance**

Liste d'indicateurs pour lesquels chaque utilisateur désire faire un suivi. Si la notification est activée dans Metric Studio, l'utilisateur recevra une notification des modifications apportées à ces indicateurs par courrier électronique. Les utilisateurs peuvent aussi afficher leur liste de surveillance sous forme de portlet dans Cognos Connection.

---

## **M**

### **magasin d'indicateurs**

Base de données qui renferme du contenu pour les packs d'indicateurs. Un magasin d'indicateurs contient aussi les paramètres de Metric Studio, tels que les préférences de l'utilisateur.

**MDX** Voir langage d'expression multidimensionnelle.

### **membre**

Élément unique dans une hiérarchie. Par

exemple, Matériel de camping et Tente à quatre personnes sont des membres de la hiérarchie Produits.

#### **membre calculé**

Membre d'une dimension dont les valeurs de mesures ne sont pas stockées, mais calculées lors de l'exécution à l'aide d'une expression.

#### **mesure**

Indicateur de performance quantifiable qui sert à déterminer comment se comporte une société. Par exemple, les mesures peuvent être Revenus, Revenus/Employé et pourcentage de marge bénéficiaire.

#### **miniature**

Rendu sous forme d'icône d'une image graphique de plus grandes dimensions, qui permet à un utilisateur de prévisualiser l'image sans ouvrir de vue ni d'éditeur graphique.

#### **modèle**

Représentation physique ou métier de la structure de données provenant d'une ou de plusieurs sources de données. Un modèle décrit des objets de données, la structure, le regroupement ainsi que les relations et la sécurité. Dans Cognos BI, un modèle est créé et géré dans Framework Manager. Le modèle ou un sous-ensemble du modèle doit être publié sur le serveur Cognos sous la forme d'un pack pour que les utilisateur puissent créer et exécuter des rapports.

#### **modèle**

Dans le cadre de la création de rapport, présentation ou style de rapport réutilisable qui peut servir dans la définition de la présentation d'une requête ou d'un rapport.

---

## **N**

#### **niveau**

Ensemble d'entités ou de membres qui forme une section de hiérarchie dans une dimension et représente le même type d'objet. Par exemple, une dimension géographique peut contenir des niveaux pour les régions, les états et les villes.

#### **normalisation**

Processus de restructuration d'un modèle de données qui vise à réduire ses

relations à leur plus simple expression. Il s'agit d'une étape majeure dans la phase de conception d'une base de données relationnelle logique. La normalisation permet d'éviter les redondances et les incohérences entre les données. Une entité est normalisée lorsqu'elle respecte un ensemble de contraintes pour un formulaire normal spécifique (premier formulaire normal, deuxième formulaire normal et ainsi de suite).

---

## **O**

**objet** Dans Report Studio, un objet est un conteneur d'informations qu'il est possible de faire glisser vers un rapport depuis l'onglet Boîte à outils, puis remplir à l'aide de données. Les rapports sont constitués d'objets, notamment des tableaux croisés, des éléments texte, des calculs, des graphiques et des tables.

---

## **P**

**pack** Sous-ensemble d'un modèle, qui peut être le modèle en entier, à rendre disponible sur le serveur Cognos. Voir aussi pack d'indicateurs.

#### **pack d'indicateurs**

Dans Cognos Connection, représentation d'une application Metric Studio. Un pack d'indicateurs contient des informations de connexion, des rapports et des tâches de gestion des indicateurs pour cette application. Voir aussi pack.

#### **paramètres régionaux**

Paramètre qui identifie la langue ou la situation géographique et détermine les conventions de formatage telles que le classement, la conversion de la casse, le classement des caractères, la langue des messages, la présentation de la date et de l'heure, ainsi que le format des nombres.

#### **paramètres régionaux de contenu**

Code utilisé pour définir la langue ou le dialecte à utiliser dans le navigateur, le contenu d'un rapport, ainsi que les préférences régionales, notamment le format de la date, de l'heure et de la devise.

#### **passeport**

Informations de session, enregistrées et chiffrées dans la mémoire de Content

Manager, relatives aux utilisateurs authentifiés. Un passeport est créé la première fois qu'un utilisateur accède à Cognos 8, il sera conservé jusqu'à la fin de la session, soit lorsque l'utilisateur se déconnecte, soit après une période d'inactivité donnée.

#### **passer au niveau inférieur**

Dans une représentation multidimensionnelle des données, accès aux informations en commençant par une catégorie générale puis en descendant à travers la hiérarchie des informations, par exemple, l'accès aux trimestres et aux mois à partir des années.

#### **passerelle**

Extension d'un programme de serveur Web qui transfère les informations depuis le serveur Web vers un autre serveur. Les passerelles sont souvent des programmes CGI, mais peuvent suivre d'autres normes, telles que ISAPI et Apache.

#### **portlet**

Composant réutilisable faisant partie d'une application Web et qui fournit des informations ou services spécifiques destinées à être présentées dans le contexte d'un portail.

#### **présentation**

Disposition des éléments imprimés sur un écran ou une page, tels que les marges, l'espacement entre les lignes, les indications de composition, les informations relatives aux en-têtes et aux pieds de page, entre autres.

#### **projet**

1. Dans Metric Studio, tâche ou ensemble de tâches pris en charge par une équipe et dont le suivi est effectué à l'aide d'un scorecard. Un projet contient des informations sur les dates, les ressources et le statut.
2. Dans Framework Manager, ensemble de modèles, de packs et d'informations connexes destiné à l'administration et au partage d'informations de modèle.
3. Dans Metric Designer, il s'agit d'un groupe d'extraits. Chaque extrait contient les métadonnées qui sont utilisées pour alimenter le magasin de données de Metric Studio ou pour créer des applications.

#### **publication**

Dans Cognos BI, exposition complète ou partielle d'un modèle Framework Manager ou d'un PowerCube Transformer à l'aide d'un pack sur le serveur Cognos afin de permettre aux utilisateurs d'en exploiter les données pour créer des rapports ou tout autre type de contenu.

---

## **R**

#### **rapport**

Ensemble de données préparé pour communiquer des informations métier.

#### **récapitulatif**

En matière de reporting et d'analyse de données, valeur agrégée calculée pour toutes les valeurs d'un niveau ou d'une dimension spécifique. La somme, le minimum, le maximum, la moyenne et le nombre sont des exemples de récapitulatifs.

#### **règle de surveillance**

Condition définie par l'utilisateur qui détermine si un rapport est envoyé à l'utilisateur. Lorsque la règle est exécutée, la sortie est évaluée et, si la condition ou la règle est satisfaite, le rapport est envoyé par courrier électronique ou sous forme d'article de forum. Les règles de surveillance limitent l'envoi de rapports aux rapports contenant des données significatives pour l'utilisateur.

#### **règle d'exécution des tâches**

Option, définie par l'utilisateur dans un agent, qui détermine les statuts et les valeurs qui entraînent l'exécution d'une tâche. Elle détermine les tâches à exécuter pour chaque instance de l'événement.

#### **répéteur**

Dans Report Studio, conteneur de cellules dans lequel des valeurs sans structure interne prédéfinie sont copiées.

#### **requête**

Spécifications de rapport simples créées et éditées par Query Studio.

#### **restriction**

Ensemble de règles qui limitent les activités des utilisateurs, notamment lorsque l'exécution d'un rapport prend trop de temps ou utilise trop de ressources.

## RSS

1. Voir Really Simple Syndication.
2. Voir RSS (Rich Site Summary).

### RSS (Really Simple Syndication)

Format de fichier XML pour le contenu Web syndiqué qui repose sur la spécification RSS 2.0. Les formats de fichier RSS XML permettent aux internautes de s'abonner aux sites Web offrant des flux RSS. Voir aussi RSS (Rich Site Summary).

### RSS (Rich Site Summary)

Format XML destiné aux contenus Web syndiqués et reposant sur la spécification RSS 0.91. Les formats de fichier RSS XML permettent aux utilisateurs Internet de s'abonner à des sites Web incluant des flux RSS. Voir aussi RSS (Really Simple Syndication).

---

## S

**score** Nombre ou classement qui exprime le respect des conditions d'application d'une norme.

### scorecard

Collection d'indicateurs représentant la performance d'une unité ou d'un aspect d'une organisation.

### scorecard de responsabilisation

Scorecard créé automatiquement par Metric Studio, pour chaque utilisateur, contenant tous les indicateurs et projets dont ils sont propriétaires.

### segment de modèle

Partie d'un projet Framework Manager, telle qu'une table de mappages des paramètres, une source de données, un espace-noms ou un dossier, qui sert de raccourci vers un autre projet. Les segments servent à simplifier la maintenance du modèle ou à faciliter la modélisation impliquant plusieurs utilisateurs.

### session

La période au cours de laquelle un utilisateur authentifié est connecté.

### sortie de rapport

Sortie produite par l'exécution d'une spécification de rapport à partir d'un ensemble de données.

### source de données

La source de données elle-même, telle qu'une base de données ou un fichier XML, ainsi que les informations de connexion nécessaires à l'accès aux données.

### source de données dimensionnelles

Source de données contenant des données modélisées à l'aide de concepts OLAP, telles que les dimensions, hiérarchies et mesures.

### source de données multidimensionnelles

Voir source de données dimensionnelles.

### sous-fenêtre d'informations

Dans Analysis Studio, une sous-fenêtre qui permet à l'utilisateur de confirmer sa sélection dans l'arborescence de données en affichant les informations associées telles que le niveau et les attributs.

### sous-fenêtre Propriétés

Dans un studio, sous-fenêtre qui fournit un aperçu des propriétés des données sélectionnées. La sous-fenêtre Propriétés peut également servir à apporter des modifications et à les appliquer simultanément, plutôt que de répéter plusieurs commandes différentes.

### spécification de déploiement

Définition des packs à déplacer (déployer) entre les environnements source et cible, les préférences de déploiement et le nom de l'archive. Les spécifications de déploiement sont utilisées pour l'importation et l'exportation.

### spécification de rapport

Définition exécutable d'un rapport, incluant les règles de requêtes et de présentation, qui peut être combinée aux données pour produire une sortie de rapport.

### stratégie

Plan d'action global (pour une marque, une unité commerciale, un réseau de distribution ou une société, par exemple) permettant d'atteindre un objectif donné. Les stratégies sont généralement établies sur une période supérieure à un an.

### structure

Zone d'un tableau de bord ou d'un espace de travail qui permet aux utilisateurs

d'interagir avec celui-ci afin de créer, visualiser et manipuler un contenu et des données.

#### **structure de scorecard**

Hiérarchie des scorecards qui reflète comment la société organise ses indicateurs.

#### **style de classe**

Combinaison de caractéristiques de formatage, telles que la police, la taille de la police et les bordures, que l'utilisateur nomme et stocke sous forme d'ensemble.

#### **sujet de requête**

Ensemble nommé d'éléments de requêtes dont les fonctionnalités sont proches. Les sujets de requêtes sont définis à l'aide de Framework Manager pour représenter des données relationnelles et former l'ensemble des données disponibles pour la création de rapports dans Query Studio et Report Studio. Le sujet de requête est comparable à une vue relationnelle car il peut être traité comme une table, mais il ne reflète pas nécessairement le stockage des données.

---

## **T**

#### **tableau de bord**

Page Web pouvant contenir un ou plusieurs composants graphiques qui représentent graphiquement des données professionnelles.

#### **table de répéteur**

Dans Report Studio, conteneur de type Table qui copie des cellules vers la droite et vers le bas dans la page ou la ligne de la requête associée.

**tâche** Action exécutée par un agent si le statut d'événement satisfait les règles d'exécution des tâches. Par exemple, un agent peut envoyer courrier électronique, publier un article de forum ou produire un rapport.

**travail** Groupe d'objets exécutables, tels que des rapports, agents et autres travaux que l'utilisateur programme et exécute par lots.

**tuple** Ensemble ordonné de plusieurs membres provenant de différentes dimensions. Par exemple, le tuple (2007, Matériel de camping, Japon) renvoie la valeur de

l'intersection des trois membres : 2007, Matériel de camping et Japon. Les tuples peuvent servir à filtrer et trier les données, ainsi qu'à créer des calculs.

#### **type d'indicateur**

Catégorie d'indicateurs qui définit des règles métier, telles que les motifs de performance, les unités et la signification d'un groupe d'indicateurs. Par exemple, Revenus peut être un type d'indicateur et Revenus - Europe et Revenus - Amérique du Nord des indicateurs de ce type.

---

## **U**

#### **utilisateur**

Tout individu, organisation, processus, périphérique, programme, protocole ou système qui utilise les services d'un système informatique.

---

## **V**

#### **vue de rapport**

Référence à un autre rapport ayant ses propres propriétés, telles que des valeurs d'invite, des plannings et des résultats. Les vues de rapport permettent de partager une spécification de rapport, plutôt que d'en faire des copies.

---

## **W**

#### **widget**

Application ou portion de contenu portable et réutilisable qui peut être placée sur une page Web, recevoir des données d'entrée et communiquer avec une application ou avec un autre widget.

#### **WSRP (Web Services for Remote Portlets)**

Norme de création de services Web de présentation pour qu'ils s'intègrent facilement aux autres applications, telles que les portails Web.

---

## **Z**

#### **zone de travail**

Zone dans un studio qui contient le rapport, l'analyse, la requête ou l'agent actuellement utilisé.





---

# Index

## A

accès  
  connexions de source de données 43  
  InfoCube sécurisé 198  
accès administrateur 266  
accès administrateur à des packs  
  modification 266  
accès au service à l'adresse URL impossible 395  
accolades 165  
accolades dans les expressions 397  
additive 141, 226  
Advisor, Model 190  
affichage  
  diagrammes 15, 28, 29  
  inclusion dans des packs 276  
  objets associés 89, 121, 214  
  objets de modèle 11, 27  
  propriétés 17  
  récapitulatifs 19  
  SQL 99  
  statistiques 19  
  tâches 19  
affichage de dépendances 293  
afficheur de projet 11  
agrégation 331  
  règles pour les métadonnées relationnelles 136, 139, 141  
  règles pour les métadonnées SAP BW 224, 226  
  règles pour métadonnées SAP BW 227  
  types 226  
  types pour les métadonnées relationnelles 141  
agrégation de calculs 344  
agrégations calculées  
  métadonnées relationnelles 139  
agrégations récapitulatives  
  métadonnées relationnelles 140  
ajout  
  ensembles de fonctions 321  
  fonctions de source de données 321  
  groupes 260  
  langues à des packs 266  
  langues pour métadonnées relationnelles 129  
  règles métier pour métadonnées relationnelles 151  
  règles métier pour métadonnées SAP BW 240  
  rôles 260  
  sécurité 266  
  sécurité des métadonnées 266  
  sécurité des objets 263  
  utilisateurs 260  
alias  
  utilisation à l'aide de paramètres 160, 246  
amélioration des performances  
  réutilisation des données mises en cache 318  
analyse  
  impact de la publication sur les packs 292  
  modèles 190  
  problèmes 3  
Analysis Studio et données fragmentées 72  
apostrophes 165  
apostrophes et guillemets 165  
  dans les expressions 165  
  dans les macros 165

application  
  filtres pour métadonnées relationnelles 157  
  filtres pour métadonnées SAP BW 244  
arborescence d'explorateur 27  
assistant de métadonnées 32  
Aster Data 2  
attributs 106, 107, 116, 117, 205, 211, 213, 387  
audit 276, 292

## B

base de données seulement 317  
bases de données  
  importation 47  
BLOB 310  
BmtScriptPlayer  
  syntaxe 301  
branchement  
  méthodologies 278  
  projets 277  
branches  
  création 280

## C

calculs  
  création pour métadonnées relationnelles 152  
  création pour métadonnées SAP BW 240  
  ordre des opérations 344  
  sécurité 264  
calculs,  
  différences inexplicables dans les nombres 399  
calculs de nombres inexplicables 399  
caractères japonais  
  chemins d'exportation 398  
caractéristique  
  mappage sur des objets Framework Manager 203  
cardinalité  
  1-1 370  
  1-n 370  
  dimensions et faits 349  
  faits 190  
  mixte 190  
  notation 70  
  redéfinition 73  
  règles 71, 328  
  requêtes 71, 328  
  source de données 69  
  types 71, 328  
  vérification 349  
cardinalité facultative 71, 328  
cardinalité maximale 71, 328  
cardinalité minimale 71, 328  
cardinalité obligatoire 71  
catalogue 324  
catégories  
  pour vérification 256  
chaînes imbriquées 165  
  dans les expressions 165  
  dans les macros 165

- chaînes littérales 165
  - dans les expressions 165
  - dans les macros 165
- changement de nom
  - projets 290
  - relations 73
- chemins ambigus 179
- chiffres clés 198
  - invites 398
  - mappage sur des objets Framework Manager 203
  - modification 214
  - structures 199
- chiffres clés calculés 198
- chiffres clés de requête SAP BW
  - externalisation 662
- chiffres clés détaillés 653
- chiffres clés InfoCube
  - externalisation 662
- choix
  - traitement de requête 317
- clause With 309
- clés 107, 205, 387
  - cardinalité 69
  - définir des rôles 116, 117
  - externes 75
  - pour les rôles 211, 213
  - pour niveaux 114
  - primaires 75
- clés externes 75
- clés primaires 75
- clés uniques 387
- Cognos PowerCubes
  - Voir* IBM Cognos PowerCubes
- commentaires
  - ajout au code SQL 99
- Common Warehouse Metamodel 295
  - exportation 398
- comparaison de tables 89
- compréhension
  - métadonnées 275
- concepts 327
- concepts de modélisation relationnelle 327
- concepts de schéma en étoile 356
- connexion à la base de données,
  - Voir* connexions de source de données
- connexion à plusieurs PowerCubes 40
- connexions de source de données 43
  - accès 43
  - création 43
  - cubes OLAP 39
  - Microsoft SQL 43
  - niveaux d'isolement 40
  - test 45
- connexions entre
  - dimensions 121, 214
  - sujets de requête 89
- Context Explorer 89, 121, 214
- contrôle
  - accès 260
  - génération SQL 309
- conventions de dénomination
  - objets 21
- conversion
  - dimensions ordinaires 125
  - éléments de requête en mesures 151
  - mesures en éléments de requête 118, 119
- conversion (*suite*)
  - sujets de requête de modèle en sujets de requête de source de données 98
  - sujets de requête en dimensions 98
- copie
  - projets 289
- correction
  - transactions ayant échoué 283
- couche de base de données 67
- couche métier 67
- couche physique 67
- couche présentation 67
- couches 67
- création 295
  - branches 280
  - calculs pour métadonnées relationnelles 152
  - calculs pour métadonnées SAP BW 240
  - connexions de source de données 43
  - dimensions de mesure 360
  - dimensions de mesure pour métadonnées relationnelles 118
  - dimensions ordinaires 358
  - dimensions ordinaires pour métadonnées relationnelles 108, 121
  - dimensions pour métadonnées relationnelles 106, 107, 116, 117
  - dimensions pour métadonnées SAP BW 205
  - dossiers d'élément de requête pour métadonnées relationnelles 186
  - dossiers de mesures pour métadonnées relationnelles 187
  - dossiers pour métadonnées relationnelles 185
  - dossiers pour métadonnées SAP BW 252
  - ensembles de requêtes pour métadonnées relationnelles 89
  - espaces-noms 65
  - espaces-noms pour métadonnées relationnelles 185
  - espaces-noms pour métadonnées SAP BW 252
  - filtres pour métadonnées relationnelles 155
  - filtres pour métadonnées SAP BW 242
  - groupes sous forme de schémas en étoile 361
  - invites avec des macros de requêtes 169
  - liens 287
  - packs 259
  - projets 9, 32
  - raccourcis de relation pour métadonnées relationnelles 74
  - relations pour métadonnées relationnelles 74
  - segments 286
  - sujets de requête pour métadonnées relationnelles 77, 79, 83
  - sujets de requête pour métadonnées SAP BW 219
  - tables de mappage des paramètres pour métadonnées relationnelles 160
  - tables de mappage des paramètres pour métadonnées SAP BW 246
- crochets 165
- CSVIdentityName, fonction 262
- CSVIdentityNameList, fonction 263
- cubes 324
- cubes OLAP
  - en tant que sources de données 39
- cumuls d'agrégat 317
- CWM
  - Voir* Common Warehouse Metamodel

## D

- date
  - invites 147, 230
  - type de format 145, 227
- DB2 397
- DB2 Cube Views
  - expressions importées 64
- DecisionStream
  - importation 49
- définition
  - contrôles d'invite 146
  - contrôles d'invite pour métadonnées SAP BW 228
  - ensembles de fonctions 321
  - langues pour métadonnées relationnelles 129
  - restrictions 309
  - suppression 267
- dépendances, objet 293
- dépendances, rapport 292
- déplacement
  - métadonnées 289
  - projets 290
- désélection
  - sécurité des objets 263
- détection
  - relations 75
- déterminants 84
  - cardinalité 69
  - conversion à partir d'informations de dimension 387
  - définition 331
  - génération SQL 88
  - identifiés de manière unique 84, 87
  - relations 190
  - spécification 84, 87
  - sujets de requête 341
- déterminants identifiés de manière unique 84, 87
- devise
  - type de format 145, 227
- Diagramme, onglet 15
- diagramme d'explorateur 28
- diagramme de sujet de requête 29
- diagrammes 11
  - affichage 15, 28, 29
  - paramètres 16, 31
- dimensions
  - ambiguïté 376
  - cardinalité mixte 190
  - conversion en sujets de requête 387
  - création 17
  - de mesure 346
  - de rôles 350
  - définition 8
  - exploration 15, 28
  - groupes sous forme de schémas en étoile 361
  - hiérarchies 358
  - identification 349
  - mesure 360
  - mesures semi-additives 143
  - modèle 335
  - modification 17
  - ordinaires 335, 341, 346, 358
  - partagées 346
  - propriétés 17
  - recherche 19
  - SAP BW 653
  - sujets de requête 335
- dimensions conformes 362
  - création 89
  - dimensions conformes (*suite*)
    - faits multiples 367, 372
- dimensions de caractéristique
  - externalisation 662
- dimensions de mesure 346
  - création 360
  - de rôles 350
  - externalisation à partir de SAP BW 662
- dimensions de mesure pour métadonnées relationnelles
  - création 118
  - exploration 121
  - relations de portée 119, 120
- dimensions de mesure pour métadonnées SAP BW
  - exploration 214
- dimensions de rôles 190, 350
- dimensions en étoile 181
- dimensions en flocons 181
- dimensions externalisées
  - invites 398
- dimensions ordinaires 335, 341, 346
  - création 358
  - de rôles 350
  - hiérarchies 358
- dimensions ordinaires pour métadonnées relationnelles 108
  - conversion 125
  - exploration 121
  - fusion 121
  - hiérarchies 110
  - niveaux 110, 113
  - relations de portée 119, 120
- dimensions ordinaires pour métadonnées SAP BW 205
  - exploration 214
  - hiérarchies 206
  - niveaux 206, 211
- dimensions partagées 346
- dimensions pour métadonnées relationnelles 106, 177
  - clés 114
  - conversion 125
  - conversion à partir de sujets de requête 98
  - en flocons 181
  - exploration 121
  - fusion 121
  - grille des dimensions 106, 107, 116, 117
  - hiérarchies 110
  - hiérarchies de réseau 113
  - hiérarchies équilibrées 112
  - hiérarchies irrégulières 113
  - hiérarchies non équilibrées 112
  - mesure 118
  - modification 18
  - niveaux 110, 113
  - ordinaires 108
  - raccourcis 183
  - relations de portée 119, 120
  - rôles 116, 117
  - test 122
- dimensions pour métadonnées SAP BW 205
  - chiffres clés 214
  - grille des dimensions 205
  - hiérarchies 206
  - hiérarchies de réseaux 211
  - hiérarchies équilibrées 207
  - hiérarchies irrégulières 209
  - hiérarchies non équilibrées 208
  - niveaux 206, 211
  - ordinaires 205
  - raccourcis 251

- dimensions pour métadonnées SAP BW (*suite*)
  - rôles 211, 213
  - test 215
- dimensions SAP BW 653
- documentation du modèle 276
- données
  - formatage pour métadonnées relationnelles 145
  - formatage pour métadonnées SAP BW 227
  - sécurité 260, 262
  - utilisation multilingue 125
- données de fait 357
- données dimensionnelles 356
- données fragmentées
  - mesures semi-additives 143
- données fragmentées et Analysis Studio 72
- données mises en cache 309
  - réutilisation 318
- données SAP BW
  - Cognos Planning 653
- dossiers
  - création 185, 187
  - de mesures 187
  - élément de requête 186
  - utilisation de raccourcis 181
- dossiers de mesures 187
- dossiers pour métadonnées SAP BW
  - création 252
  - utilisation de raccourcis 251
- double comptage 84, 349, 370, 376

## E

- éditeur d'expression
  - Aster Data 447
  - chaîne d'Aster DATA 447
  - Chaîne Greenplum 470
  - Chaîne MySQL 489
  - Chaîne Paracel 509
  - Chaîne Postgres 511
  - Chaîne Vectorwise 548
  - Chaîne Vertica 551
  - Constantes 433
  - DB2 451
  - ensembles de fonctions 321
  - fonctions communes 573
  - fonctions de bloc 445
  - fonctions de conversion MS Access 481
  - fonctions de date 523
  - fonctions de date-heure selon le principe des jours ouvrables 436
  - fonctions de rapport 612
  - fonctions dimensionnelles 585
  - fonctions macros 556
  - fonctions mathématiques DB2 451
  - fonctions mathématiques Greenplum 472
  - fonctions mathématiques Informix 475
  - fonctions mathématiques MS Access 482
  - fonctions mathématiques MySQL 490
  - fonctions mathématiques Netezza 493
  - fonctions mathématiques Oracle 499
  - fonctions mathématiques Paracel 510
  - fonctions mathématiques Postgres 513
  - fonctions mathématiques SAP BW 523
  - fonctions mathématiques SQL Server 527
  - fonctions mathématiques Sybase 534
  - fonctions mathématiques Vectorwise 549
  - fonctions mathématiques Vertica 553
- éditeur d'expression (*suite*)
  - fonctions phonétiques Netezza 494
  - fonctions statistiques 412
  - fonctions trigonométriques 583
  - fonctions trigonométriques DB2 452
  - fonctions trigonométriques Greenplum 473
  - fonctions trigonométriques Informix 476
  - fonctions trigonométriques MS Access 482
  - fonctions trigonométriques MySQL 491
  - fonctions trigonométriques Netezza 493
  - fonctions trigonométriques Oracle 499
  - fonctions trigonométriques Postgres 514
  - fonctions trigonométriques SAP BW 521
  - fonctions trigonométriques SQL Server 527
  - fonctions trigonométriques Sybase 535
  - fonctions trigonométriques Teradata 543
  - fonctions trigonométriques Vectorwise 549
  - fonctions trigonométriques Vertica 553
  - formatage des types de données Paracel 509
  - formatage des types de données Postgres 512
  - formatage des types de données Vertica 552
  - formatage du type de données Aster DATA 448
  - formatage du type de données Greenplum 472
  - Greenplum 470
  - Informix 475
  - Mathématiques appliquées à Aster DATA 449
  - MS Access 481
  - MySQL 489
  - Netezza 493
  - Netezza Fuzzy 494
  - Opérateurs 403
  - Oracle 499
  - Paracel 509
  - Postgres 511
  - Récapitulatifs 411
  - Récapitulatifs des membres 430
  - recherche de valeurs 401
  - Red Brick 516
  - Salesforce.com 523
  - SAP BW 521
  - SQL Server 527
  - Sybase 534
  - Syntaxes 435
  - Teradata 543
  - Trigonométrie sur Aster DATA 449
  - Vectorwise 548
  - Vertica 551
- édition
  - SQL 99
- éléments de requête
  - conversion 119, 151
  - conversion à partir de mesures 118
  - définition 8
  - propriétés 17
- éléments de requête pour métadonnées relationnelles
  - changement de nom 292
  - dossiers 186
  - fonction de modèle durable 292
  - modification 18, 132
  - règles d'agrégation 141
  - rôles 116, 117
  - usage 140
- éléments de requête pour métadonnées SAP BW
  - formatage 227
  - modification 220
  - rôles 213

- enregistrement automatique
  - options 13
- enregistrement des transactions 299
- enregistrer sous
  - copie de projets 289
- ensemble de données par défaut
  - définition 297
- ensemble de requêtes (opération Except) 89
- ensemble de requêtes (opération Except All) 89
- ensemble de requêtes (opération Intersect) 89
- ensemble de requêtes (opération Intersect All) 89
- ensemble de requêtes (opération Union) 89
- ensemble de requêtes (opération Union All) 89
- ensembles de données multiples 296
  - définition d'ensembles de données par défaut 297
- ensembles de fonctions
  - ajout 259, 321
  - définition 321
- ensembles de requêtes pour métadonnées relationnelles
  - création 89
  - test 93
- entrepôt de données
  - relations 68
- environnement
  - Framework Manager 5
- environnement de génération de rapports multilingues
  - configuration 125
- environnements pris en charge
  - multilingue 125
- erreur de mémoire insuffisante 395
- erreurs
  - réparation 256
- erreurs d'arrondi 399
- erreurs d'arrondi binaire 399
- espaces-noms 5
  - création 65
  - création pour métadonnées relationnelles 185
  - création pour métadonnées SAP BW 252
  - définition 7
- évaluation
  - sujets de requête relationnelle 96
  - sujets de requête SAP BW 220
- Event Studio
  - procédures stockées 83
- exécuter à nouveau
  - journaux des actions 301
- exemples de modèle 23
- exemples pour métadonnées relationnelles
  - filtre 159
  - in\_range, fonction 155
  - invites 171
  - modélisation multilingue 131
  - table de mappage des paramètres 162
- exemples pour métadonnées SAP BW
  - in\_range, fonction 242
- exigences de génération de rapports 3
- exploration
  - dimensions 106, 121, 205, 214
  - objets 15, 28
  - packs 275
  - rôles 275
  - sécurité orientée objets 275
  - sujets de requête 89
- exportation
  - fichier CWM 398
  - métadonnées en tant que fichiers CWM 295
  - tables de traduction 130

- expressions
  - accolades 397
  - importées 64
  - personnalisation 73
  - relations 73
  - utilisation d'invites 146
- expressions complexes
  - relations 73
- expressions de relation personnalisée 73
- externalisation
  - chiffres clés de requête SAP BW 662, 665
  - chiffres clés InfoCube 662
  - dimensions de caractéristique 662
  - dimensions de mesure 662
  - dimensions SAP BW 653
  - exigences relatives à Framework Manager 664
  - exigences relatives aux requêtes SAP BW 663
  - sujets de requête 267, 270
  - sujets de requête de modèle 662
- extraction de données
  - instructions SAP BW pour l'utilisation de variables 656

## F

- faits 360
  - ambiguïté 376
  - cardinalité 190
  - cardinalité mixte 190
  - identification 349
  - règles d'agrégation 143
- fichier journal
  - archiver des entrées 305
- fichiers CSV 267, 270
- fichiers de définition de requête Impromptu 270
- fichiers imbriqués 267, 270
- fichiers journaux 299
  - exécuter à nouveau 300
- fichiers script
  - correction des erreurs 304
- fichiers TAB 267, 270
- fichiers XML
  - importation 65
- fichiers XML Architect
  - importation 49
- fichiers XML Impromptu
  - importation 49
- filtres
  - facultatifs 157
  - obligatoires 157
  - sécurité 262, 264
- filtres pour métadonnées relationnelles
  - application aux sujets de requête 157
  - création 155
  - détermination de l'utilisation 157
  - dimensions multiples 18
  - éléments de requête multiples 18
  - facultatifs 155
  - modification 155
  - obligatoires 155
  - sujets de requête multiples 18
- filtres pour métadonnées SAP BW
  - application 244
  - création 242
  - détermination de l'utilisation 244
  - modification 242
- fonctions 2
  - modifiées 1

- fonctions (*suite*)
  - nouvelles 1
  - obsolètes 1
  - projet 321
  - supprimées 1
- fonctions d'accessibilité 393
  - raccourcis-clavier 393
- fonctions de base de données
  - spécifiques à des fournisseurs 321
- fonctions de base de données spécifiques à des fournisseurs 321
- fonctions de source de données
  - ajout 321
- fonctions modifiées 1
- fonctions obsolètes 1
- fonctions supprimées 1
- formatage
  - données pour métadonnées relationnelles 145
  - données pour métadonnées SAP BW 227
- formats de données
  - à l'aide de motifs 641
  - symboles de date et d'heure 642
  - symboles des décimales 650
- Framework Manager
  - projet 5
- fusion
  - dimensions ordinaires relationnelles 121
  - projets 277
  - sujets de requête, SAP BW 220
  - sujets de requête relationnels 88

## G

- génération
  - modèles, à l'aide de Model Design Accelerator 34
  - relations 75
- génération SQL
  - contrôle 309
- génération SQL pour les attributs de déterminant
  - restriction 313
- génération SQL pour les attributs de niveau
  - restriction 312
- gestion
  - packs 263, 266, 275, 276
  - projet 275
  - projets 289, 299
- gestion des versions
  - modèles 270
- gestion des versions de modèle 274
- glossaire 677
- granularité 354
- graphic, type de données 396
- grille des dimensions 106, 107, 116, 117, 205
- Grille des dimensions, onglet 17
- groupes
  - ajout 260
  - schéma en étoile 177
- groupes sous forme de schémas en étoile 346
  - conformes multiples 362
  - création 361
- groupes sous forme de schémas en étoile conformes 362
- guillemets 165

## H

- hiérarchies 331, 335, 387
  - avec contrainte horaire 201
  - dimension ordinaire 110, 206
  - équilibrées 112, 207
  - irrégulières 113, 209
  - non équilibrées 112, 208
  - plusieurs 358
  - réseau 113, 211
  - SAP BW 200
  - spécification de rôles 116, 117
  - versionnées 201
- hiérarchies de réseau 113
- hiérarchies de réseaux 211
- hiérarchies équilibrées 112, 207
- hiérarchies irrégulières 113, 209
- hiérarchies non équilibrées 112, 208
- hiérarchies versionnées 201
- historique des transactions
  - affichage 300

## I

- IBM Cognos 8 Planning et données SAP BW 653
- IBM Cognos PowerCubes
  - publication 272
- ID d'ensemble de fonctions 324
- identificateurs
  - uniques 84
- identificateurs uniques 84
  - objets 21
- importation
  - bases de données 47
  - DecisionStream 49
  - fichiers XML Architect 49
  - fichiers XML Impromptu 49
  - hiérarchies SAP BW 200
  - métadonnées dans Framework Manager 46
  - métadonnées issues de fichiers XML 65
  - métadonnées relationnelles 75
  - métadonnées SAP BW 198, 202
  - modèles IBM Cognos 48
  - noms d'objet en double 46
  - relations 68
  - sources de données 57
  - tables de traduction pour métadonnées relationnelles 130
- in\_range, fonction 155, 242
- InfoCube
  - autorisations d'accès aux métadonnées 198
  - mappage sur des objets Framework Manager 203
- InfoCube sécurisé
  - accès 198
- informations de dimension 387
- Informix
  - fonctions 81
  - procédures stockées 81
- internes, jointures 399
- invite
  - filtrer par référence 149, 232
- invites
  - afficher la référence d'élément 148, 231
  - cascade sur la référence 148, 231
  - chiffres clés 398
  - création 169
  - création avec des macros 169
  - dans les expressions 146

- invites (*suite*)
  - date 146, 147, 228, 230
  - définition 146, 228
  - différentes valeurs à utiliser et à afficher 148, 231
  - exemple de procédure stockée 84
  - exemples 171
  - facultatives 169
  - heure 146, 147, 228, 230
  - liste déroulante 146, 147, 228, 230
  - obligatoires 169
  - propriétés 146, 147, 228, 230
  - recherche 146, 147, 228, 230
  - saisie au clavier 146, 147, 228, 230
  - test 150, 233
  - utiliser la référence d'élément 148, 231
  - variables SAP BW 234
  - zone de texte 147, 228, 230
- invites de liste déroulante 147, 230
- invites de recherche 147, 230
- invites de saisie au clavier 146, 147, 228, 230
- invites de saisie d'heure 146, 228
- invites de saisie d'une heure 147, 230
- invites de saisie de date 146, 228
- invites de saisie de liste déroulante 146, 228
- invites de saisie de recherche 146, 228
- invites de saisie de zone de texte 146, 228
- invites de zone de texte 147, 230
- invites facultatives 169
- invites obligatoires 169
- IQD
  - fichiers 270

## J

- jointure externe complète 69, 396
- jointures 68
  - boucles 353
  - création 30
  - externes 70, 72
  - externes complètes 69, 396
  - internes 70, 75, 399
  - modification 30, 73
  - Oracle 399
  - remplacement 30
  - suppression 30
- jointures automatiques 190
- jointures de boucles 349, 353
- jointures externes 70, 75, 309
  - Analysis Studio 72
  - complètes 69, 396
  - Oracle 399
- jointures externes complètes
  - Oracle 399
- jointures externes complètes Oracle 396
- jointures inter-produits 309
- jointures internes 70, 75
- journaux des actions 299
  - exécution en mode différé 301

## L

- langue de l'interface utilisateur 2
- langues
  - ajout pour métadonnées relationnelles 129
  - configuration d'environnement multilingue 125
  - définition pour métadonnées relationnelles 129

- langues (*suite*)
  - incorrectes dans la requête SAP BW 396
  - spécification pour des packs 266
- liaison
  - limites 285
- liens 283
  - création 287
- Limite des éléments texte de grande taille 310
- limites
  - liaison 285
  - macros 168
  - segmentation 285
- limites de temps 309

## M

- macros
  - création 168
  - invites 165
  - limites 168
  - utilisation 165
- macros de requêtes
  - utilisation pour la création d'invites 169
- mappage
  - objets SAP BW sur des objets Framework Manager 203
  - valeurs de paramètre pour métadonnées relationnelles 160
  - valeurs de paramètre pour métadonnées SAP BW 246
- membres
  - tri 114
- message d'erreur
  - QE-DEF-0259 397
- messages d'erreur
  - QE-DEF-0177 396
  - UDA-SQL-0107 396
  - UDA-SQL-0114 396
- mesures
  - conversion 119, 151
  - conversion en éléments de requête 118
  - mise à niveau IBM Cognos BI 385
  - pour IBM Cognos 8 Planning 653
  - semi-additives 143
  - type d'agrégation calculée 190
- mesures semi-additives 143, 227
- métadonnées
  - compréhension 275
  - déplacement 289
  - exportation 295
  - fusion 88, 220
  - importation 46
  - mises en cache 190
  - modification 289
  - publication 255
  - relationnelles 67
  - SAP BW 197
  - sécurité 260, 266
  - synchronisation 305
- métadonnées DMR (relationnelles modélisées sous forme de dimension) 358
- métadonnées ERWin
  - erreur de mémoire insuffisante 395
- métadonnées importées
  - vérification 349
- métadonnées mises en cache 190
- métadonnées natives 39
- métadonnées relationnelles 67
  - éléments de requête 132

- métadonnées relationnelles (*suite*)
  - règles métier 151
  - schémas en étoile 177
  - vue d'importation 67
  - vue métier 67
- métadonnées relationnelles modélisées de façon dimensionnelle
  - mesures semi-additives 143
- métadonnées SAP BW 197
  - importation 198
- méthodologies
  - branchement 278
- Microsoft
  - sources de données SQL Server 47
  - SQL Server et connexion 43
- Microsoft Analysis Server
  - sources de données 47
- migration
  - modèles 299
- mise à jour
  - sujets de requête 97
  - versions de modèle 270
- mise à niveau
  - modèles 10, 383
  - modèles IBM Cognos ReportNet 384
  - projets liés 391
  - projets segmentés 391
- mise à niveau de modèles
  - avertissements 391
- mode d'accès de la transaction 1, 320
- mode d'édition de relation 30
- mode de l'instruction de transaction 1, 320
- mode de requête dynamique 270, 271, 309, 310, 311, 313
- Model Accelerator 29
- Model Advisor 190
- Model Design Accelerator 27
  - arborescence d'explorateur 27
  - assistant de métadonnées 32
  - couleurs du diagramme 28
  - création de projets 32
  - création de schémas en étoile 33
  - création de sujets de requête 33
  - diagramme d'explorateur 28
  - diagramme de sujet de requête 29
  - génération de modèles 34
  - gestion des schémas en étoile 35
  - jointures 30
  - mode d'édition de relation 30
  - Model Accelerator 29
  - paramètres de diagramme 31
  - vue Avertissement de modèle 31
- modèle
  - durabilité 187
  - gestion des versions 274
- modèles 5
  - analyse 190
  - définition 7
  - exemples 23
  - génération, à l'aide de Model Design Accelerator 34
  - gestion des versions 270
  - migration d'une base de données relationnelle vers une autre 299
  - mise à niveau 10, 383
  - préparation de métadonnées SAP BW 197
  - publication 255
  - relationnels, préparation 67
  - vérification 256
- modèles dimensionnels
  - schémas en étoile 177
- modèles durables 187
  - changement de nom des éléments de requête 292
- modèles IBM Cognos
  - importation 48
- modèles relationnels
  - migration 299
- modélisation
  - modèles distribués 277
  - multilingue 127
  - multiutilisateur 277
  - principaux projets 277
  - sources de données 127
- modélisation dimensionnelle de métadonnées relationnelles 358
- modélisation multilingue
  - exemple pour métadonnées relationnelles 131
  - métadonnées relationnelles 125
  - packs 266
  - paramètres 128
- modélisation multiutilisateur 277
- modification
  - accès administrateur à des packs 266
  - accès aux packs 266
  - code SQL lors de l'exécution 169
  - dimensions relationnelles 18
  - éléments de requête pour métadonnées relationnelles 132
  - éléments de requête pour métadonnées SAP BW 220
  - filtres pour métadonnées relationnelles 155
  - filtres pour métadonnées SAP BW 242
  - métadonnées 289
  - packs 259
  - paramètres de session 163, 248
  - propriétés 17
  - propriétés d'élément de requête 18
  - relations 73
  - sujets de requête de procédure stockée 83
  - sujets de requête relationnelle 18
  - type de code SQL 100
- modification de packs
  - analyse des effets 292
- modifications dans le modèle et pas dans le rapport 274
- mots réservés 669
- MultiCube
  - mappage sur des objets Framework Manager 203
- multidimensionnels
  - sujets de requête 218
- multilingue
  - sources de données 127
- multilingues
  - données, utilisation 125
- multilingues, rapports 126

## N

- niveau de détail
  - paramètres 16, 31
- niveau de hiérarchie de présentation
  - mappage sur des objets Framework Manager 203
- niveaux 106, 110, 205, 206, 211, 213, 387
  - définir des rôles 116, 117
  - dimension ordinaire 113, 211
  - noms uniques de membres 113, 211
  - sécurité 398
  - tri 114
  - uniques 108



- niveaux d'isolement 40
- niveaux uniques 108
- nombre d'éléments distincts 136
- nombre limite de lignes 309
- noms d'objet en double
  - importation 46
- noms uniques de membres
  - métadonnées relationnelles 113
  - métadonnées SAP BW 211
- non-additive 141, 226
- notation 16, 31, 69, 70
- notation Crowsfoot 16, 31
- notation Merise 16, 31
- nouveautés 1

## O

- objet CLOB
  - Oracle 395
- objets
  - conventions de dénomination 21
  - exploration 15, 28
  - identificateurs uniques 21
  - propriétés 17
  - recherche 19
  - remappage 294
  - réorganiser 15
- objets ambigus 376
- objets associés 89, 121, 214
- objets de modèle 11
  - affichage 11, 27
  - raccourcis 342
  - utilisation de paramètres 160, 246
- objets de projet
  - conventions de dénomination 21
- onglet Explorateur 15
- opérateur plus 165
- opérations de calcul 344
- optimisation des performances
  - traitement de requête 317
- options d'instantané
  - paramètres 16, 31
- options de suppression 267
- Oracle
  - Designer 63
  - jointures externes complètes 399
- ordre des opérations de calcul 344
- organisation de modèles 176, 250
- ouverture
  - projets 10

## P

- packs 5
  - accès 266
  - accès administrateur 266
  - affichage de l'inclusion 276
  - ajout de mécanismes de sécurité 266
  - analyse des modifications 292
  - contenant plusieurs cubes 272
  - création 259
  - création dans Cognos Connection 197, 270
  - définition 8
  - exploration 275
  - gestion 263, 266, 275, 276
  - modification 259

- packs (*suite*)
  - publication 255, 270
  - sécurité 260
  - support multilingue 266
  - vérification 256
- packs Oracle
  - sources de données 47
- page de projet 11
- page IBM sur l'accessibilité 395
- paramètres
  - dans des sujets de requête de source de données 165
  - niveau de détail 16, 31
  - notation 16, 31
  - options d'instantané 16, 31
  - police 16, 31
  - test 94, 123, 216
- paramètres de session
  - modification pour métadonnées relationnelles 163
  - modification pour métadonnées SAP BW 248
- paramètres de session de modèle
  - métadonnées relationnelles 163
  - métadonnées SAP BW 248
- paramètres imbriqués 160, 246
- paramètres pour métadonnées relationnelles
  - alias 160
  - création 160
  - imbriqués 160
  - modélisation multilingue 128
  - objets de modèle 160
  - session 163
  - sujets de requête de source de données 165
  - utilisation 165
- paramètres pour métadonnées SAP BW
  - création 246
  - session 248
  - tables de mappage 246
- paramètres régionaux
  - Voir aussi* langues
  - configuration d'un environnement multilingue 125
- passerelles
  - accès impossible à l'URI 395
- plusieurs connexions de source de données
  - utilisation 40, 43
- plusieurs cubes dans des packs 272
- plusieurs hiérarchies 358
- plusieurs relations valides 350
- plusieurs utilisateurs
  - modélisation 277
- police
  - paramètres 16, 31
- portabilité 299
- portabilité des modèles 299, 309
- PowerCubes
  - chemins d'accès aux fichiers de cube 44
  - connexions à plusieurs sources de données 40
  - création à partir de données SAP BW 666
  - instructions pour Transformer 666
  - instructions relatives à Framework Manager 666
- prise en charge
  - plusieurs utilisateurs 277
- problèmes de modélisation 190
- problèmes liés à la portée des mesures 119
- procédures stockées
  - requête de données mettant à jour la base de données 83
- procédures stockées Composite 81
- projet
  - gestion 275

- projets 5
  - affichage d'un historique des transactions 300
  - branchement 277
  - changement de nom 290
  - copie 289
  - correction des erreurs de synchronisation 304
  - création 9, 32
  - définition 5
  - déplacement 290
  - fichiers 5
  - fonctions 321
  - Framework Manager 5
  - fusion 277
  - gestion 289, 299
  - liaison 287
  - options 13
  - ouverture 10
  - raccourcis 287
  - réexécution de transactions 300
  - segmentation 286
  - suppression 291
  - synchronisation 305
- projets liés 283, 391
  - synchronisation 307
- projets principaux 286
- projets segmentés 283, 286, 391
  - copie 289
  - déplacement 290
  - suppression 291
  - synchronisation 306
- propriété Agrégat ordinaire 141
  - métadonnées relationnelles 136
  - métadonnées SAP BW 224, 226
- propriété Informations d'invite 146, 228
- propriété Règles d'agrégation
  - métadonnées relationnelles 136
- propriété Semi-agrégat 144
  - métadonnées SAP BW 224, 227
- propriété Usage
  - métadonnées relationnelles 136, 140
  - métadonnées SAP BW 224, 226
- propriétés
  - affichage 17
  - afficher la référence d'élément 148, 231
  - agrégat ordinaire 136, 224
  - cascade sur la référence 148, 231
  - catalogue 324
  - cubes 324
  - filtrer par référence 149, 232
  - ID d'ensemble de fonctions 324
  - Informations d'invite 146, 228
  - interface de requête 324
  - mode d'accès de la transaction 324
  - mode de l'instruction de transaction 324
  - modification 17
  - règles d'agrégation 136
  - règles d'agrégation pour les métadonnées SAP BW 224
  - schéma 324
  - semi-agrégat 224
  - source de données Content Manager 324
  - source de données SAP BW 211
  - sources de données 317
  - table de mappage des paramètres 324
  - traitement de cumul 324
  - traitement de requête 324
  - type d'invite 147, 230
  - type de requête 324

- propriétés (*suite*)
  - usage 136, 224
  - utiliser la référence d'élément 148, 231
  - variables SAP BW 211
- propriétés d'informations d'invite 146
- propriétés personnalisées
  - SAP BW 234
- publication
  - métadonnées 255
  - packs 255, 270, 273
  - packs basés sur des sources de données OLAP 272

## Q

- QE-DEF-0177 396
- QE-DEF-0259 397

## R

- raccourcis 342
  - comportement 309, 311
  - sécurité 264
- raccourcis-clavier 393
- raccourcis pour métadonnées relationnelles
  - dimensions et sujets de requête 183
  - relations 74, 183
  - utilisation 181
- raccourcis pour métadonnées SAP BW
  - dimensions 251
  - utilisation 251
- rapport
  - dépendances 292
- rapports
  - multilingues 126
  - n'affichant pas les modifications du modèle 274
- rapports altérés 273
- récapitulatifs
  - affichage 19
- recherche
  - objets 18, 19
- règles
  - agrégation pour les métadonnées SAP BW 224
  - cardinalité 71
  - métier 151
  - métier pour métadonnées SAP BW 240
- règles d'agrégation 143
- règles de cardinalité 328
- règles métier pour métadonnées relationnelles 151
  - calculs 152
  - filtres 155
  - invites 146
  - macros 165
  - paramètres 165
  - paramètres de session 163
  - tables de mappage des paramètres 160
- règles métier pour métadonnées SAP BW 240
  - calculs 240
  - filtres 242
- Regroupement des attributs de mesure
  - restriction 312
- réinitialisation
  - propriété Agrégat ordinaire 144
  - propriété Usage 144
- relations
  - 0 à 1 69
  - 0 à n 69

- relations (*suite*)
  - 1 à 1 69
  - 1 à n 69
  - 1-n 370
  - ambiguës 350
  - cardinalité 69, 328
  - création 74
  - définition 68
  - depuis Oracle Designer 63
  - détection 75
  - déterminants 190
  - expressions complexes 73
  - génération 75
  - importation 68
  - modification 73
  - multiples 190
  - multivoques 69
  - n à n 69
  - niveaux de granularité 354
  - personnalisées 73
  - plusieurs à plusieurs 69
  - plusieurs à un 69
  - plusieurs valides 350
  - portée 119, 120
  - raccourcis 74, 183
  - relations réflexives et récursives 190
  - un à plusieurs 69
  - un à un 69
  - valides 353
  - vérification 68, 349
- relations ambiguës 350
- relations de portée 106, 107, 116, 117, 119, 120, 205
- relations de type 1-1 370
- relations de type 1-n 370
- relations facultatives 69
- relations multiples 190
- relations multivoques 69
- relations plusieurs à un 69
- relations récursives 190, 354
- relations réflexives 190, 354
- relations un à plusieurs 69
- relations un à un 69
- relations valides 353
  - plusieurs 350
- remapper des objets 294
- RemoteCube
  - mappage sur des objets Framework Manager 203
- réorganiser des objets 15
- réparation
  - modèles IBM Cognos ReportNet 386
- réparation de modèles 256
- requête, SAP BW
  - mappage sur des objets Framework Manager 203
- requête sans fait 362
- requête SAP BW
  - autorisations d'accès aux métadonnées 198
- requêtes
  - à granularité multiple 335
  - fait unique 365
  - faits multiples 335, 367, 372
  - fractionnées 377
  - liées 349
  - sans fait 362
- requêtes à fait unique 365
- requêtes à faits multiples 335, 367, 372
- requêtes à granularité multiple 335, 367, 372
- requêtes de faits croisés 349

- requêtes dimensionnelles 365
  - fait unique 365
  - granularité et faits multiples 367, 372
- requêtes fractionnées 377
- requêtes liées 349
- résolution
  - objets ambigus 376
  - requêtes fractionnées 377
- restreindre
  - accès 260
  - données extraites 309
  - objets BLOB 310
- restriction
  - Autoriser l'utilisation du cache local 315
  - Autoriser la génération dynamique des informations de dimension 315
  - Autoriser la portabilité du modèle amélioré lors de l'exécution 314
  - génération SQL pour les attributs de déterminant 313
  - génération SQL pour les attributs de niveau 312
  - Jointures externes 310
  - Jointures inter-produits 311
  - Limite temporelle d'exécution de requête 310
  - Nombre maximal de lignes extraites 309
  - Nombre maximal de sources de données externes 316
  - Nombre maximal de tables de rapport 309
  - regroupement des attributs de mesure 312
  - Supprimer les valeurs nulles pour les sources de données SAP BW 315
  - Syntaxe de jointure SQL 311
  - syntaxe de paramètre SQL 314
  - Taille maximale du fichier de données externes 316
  - Traitement des raccourcis 311
  - Traitement du modèle entier lors d'une publication 316
  - Utiliser la clause WITH lors de la génération de SQL 315
- restrictions
  - autoriser l'utilisation du cache local 318
  - définition 309
  - mise à niveau d'IBM Cognos ReportNet 384
- réutilisation des données mises en cache 318
- réutilisation des requêtes 318
- rôles
  - ajout 260
  - dimensions 116, 117
  - exploration 275
- rôles pour les métadonnées SAP BW
  - dimensions 211, 213

## S

- SAP BW
  - importation de chiffres clés 198
  - instructions d'utilisation de données de faits 656
  - langue incorrecte dans la requête 396
  - propriétés personnalisées 234
  - Transformer 661
- schéma 324
- schémas en étoile
  - création à l'aide de Model Design Accelerator 33
  - gestion à l'aide de Model Design Accelerator 35
  - groupements 177
  - multiples 179
- scripts
  - exécution 273
- se connecter
  - à Microsoft SQL 43

- sécurité
  - CSVIdentityName, fonction 262
  - CSVIdentityNameList, fonction 263
  - exploration 275
  - filtres 262
  - niveaux 398
  - packs 259
  - types 260
- sécurité des objets 260
  - ajout 263
  - calculs 264
  - exploration 275
  - filtres 264
  - raccourcis 264
  - sujets de requête de modèle 264
  - suppression 263
- segmentation
  - limites 285
  - projets 286
- segments 283
  - recommandations d'utilisation 284
- semi-additive 144
- signe dièse
  - réservé pour les macros 165
- source de données
  - ajout de fonctions 321
  - mise à jour par une procédure stockée 83
- source de données Content Manager 324
- sources de données
  - autres sources de métadonnées 57
  - bases de données relationnelles 47
  - chemins d'accès aux connexions de sources de données
    - basées sur des fichiers 44
  - DecisionStream 49
  - ERwin 57
  - fichiers XML 65
  - fichiers XML Architect 49
  - fichiers XML Impromptu 49
  - IBM DB2 Cube Views 57
  - Microsoft Analysis Server 47
  - Microsoft SQL Server 47
  - modèles IBM Cognos 48
  - modification des propriétés 324
  - multilingue 127
  - niveaux de connexion 40
  - Oracle Designer 63
  - packs Oracle 47
  - propriétés 211, 317
  - SAP BW 198
- sources de données basées sur des fichiers
  - chemins d'accès 44
- sources de données en flocons 356
- sources de données normalisées 356
- sources de données OLAP
  - publication 272
- SQL 365
  - Cognos 99, 102
  - comme une vue 106
  - commentaires 99
  - déterminants 88
  - direct 99, 104
  - édition 99
  - En tant filtre 106
  - internes, jointures 399
  - modification du type 100
  - natif 99, 103
  - natif autonome 99, 104
  - SQL (suite)
    - réduit 106, 190
    - Server, connexion 43
    - sujets de requête 76
    - sujets de requête de modèle 106
    - types de génération 106
  - SQL, Comme une vue
    - sujets de requête de modèle 106
  - SQL, En tant filtre 106
  - SQL Cognos 102
  - SQL direct 104
  - SQL en saisie directe 397
  - SQL natif 103
  - SQL réduit 106, 190
    - sujets de requête de modèle 106
  - statistiques
    - affichage 19
  - structures
    - SAP BW 199
  - structures de caractéristique 199
  - structures SAP BW 199
  - sujet de requête de fait détaillé 653
  - sujets de requête
    - cardinalité mixte 190
    - conversion en dimensions 387
    - création à l'aide de Model Design Accelerator 33
    - DB2 397
    - définition 8
    - déterminants 190, 341, 387
    - dimensions 190, 335
    - exploration 15, 28
    - externalisés 267, 270
    - faits 190
    - fichiers CSV 267, 270
    - fichiers de définition de requête Impromptu 270
    - fichiers imbriqués 267, 270
    - fichiers TAB 267, 270
    - fichiers Transformer 270
    - groupes sous forme de schémas en étoile 361
    - modes de transaction 320
    - propriétés 17
    - recherche 19
    - relations multiples 190
    - SAP BW 385
    - sécurité 262
    - SQL réduit 190
    - test 63
    - sujets de requête conditionnels 160, 246
  - sujets de requête de modèle
    - définition 8
    - externalisation 662
    - sécurité 264
  - sujets de requête de modèle model pour métadonnées
    - relationnelles 78
    - création 79
    - création à partir d'objets existants 88
    - déterminants 84, 87
    - type SQL 106
  - sujets de requête de modèle pour métadonnées SAP BW 218
    - création 219
    - création à partir d'objets existants 220
  - sujets de requête de procédure stockée 80
    - Composite 81
    - création 83
    - définition 8
    - évaluation 96
    - exemple 84

- sujets de requête de procédure stockée (*suite*)
  - Informix 81
  - mise à jour 97
  - modification 83
- sujets de requête de source de données
  - création 77
  - définition 8
- sujets de requête de source de données pour des métadonnées relationnelles 76
- sujets de requête de source de données pour métadonnées relationnelles
  - déterminants 84, 87
  - utilisation de paramètres 165
- sujets de requête de source de données relationnelles
  - évaluation 96
  - mise à jour 97
- sujets de requête pour métadonnées relationnelles
  - conversion 98
  - conversion pour des dimensions 125
  - création 77, 79, 88
  - déterminants 84, 87
  - édition du code SQL 99
  - ensembles de requêtes 89
  - évaluation 96
  - exploration 89
  - génération de code SQL 106
  - mise à jour 97
  - modèle 78
  - modification 18, 83
  - modification du type de code SQL 100
  - procédure stockée 80
  - raccourcis 183
  - relations 68
  - source de données 76
  - SQL 76
  - tables de mappage des paramètres 160
  - test 93
  - types 76
- sujets de requête pour métadonnées SAP BW
  - agrégation 226
  - création 219, 220
  - évaluation 220
  - modèle 218
  - règles d'agrégation 226
  - rôles 211
  - tables de mappage des paramètres 246
  - usage 226
- sujets de requête SAP BW 385
- suppression
  - null 267
  - plusieurs arêtes 267
  - projets 291
  - sécurité des objets 263
- suppression des valeurs nulles 267
- suppression sur plusieurs arêtes 267
- symbole d'échappement 165
- symboles réservés
  - signe dièse 165
- synchronisation
  - correction des erreurs 304
  - métadonnées 305
  - projets 305
  - projets liés 307
  - projets segmentés 306
  - sujets de requête 97
- syntaxe
  - BmtScriptPlayer 301

- Syntaxe de paramètre SQL
  - restriction 314

## T

- table de faits 118
- tables de faits 177
- tables de mappage des paramètres pour métadonnées relationnelles
  - création 160
  - utilisation en tant que tables de correspondance 125
- tables de mappage des paramètres pour métadonnées SAP BW 246
  - création 246
- tables de traduction
  - exportation 130
  - importation 130
- tables Principale/Détails 357, 360
- tâches
  - affichage 19
- test
  - connexions de source de données 45
  - dimensions de mesure relationnelles 122
  - dimensions de mesure SAP BW 215
  - dimensions ordinaires relationnelles 122
  - dimensions ordinaires SAP BW 215
  - ensembles de requêtes relationnelles 93
  - invites 150, 233
  - modification des paramètres 94, 123, 216
  - projets 256
  - sujets de requête 63, 396
  - sujets de requête relationnelle 93
- trace
  - historique des transactions 300
- traitement de cumul 317, 324
- traitement de requête 317, 324
  - choix 317
- traitement de requête local
  - traitement de cumul 317
- traitement des incidents 395
- traitement local limité 317
- transactions
  - correction 283
  - exécuter à nouveau 300
  - mode 320
- Transformer
  - SAP BW 661
- Transformer et sujets de requête 270
- Transformer version 7.x
  - fichiers CSV externalisés 661
- Transformer version 8.x
  - fichiers CSV externalisés 661
- tri
  - membres 114
  - niveaux 114
  - objets 18
- trier des objets 15
- type d'agrégation
  - calculée 190
- type d'agrégation calculée 190, 344
- type de format de texte 145, 227
- type de requête 324
- types d'agrégation automatique
  - métadonnées relationnelles 139
  - métadonnées SAP BW 226
- types d'agrégation automatiques
  - métadonnées SAP BW 227

- types de données 385
  - graphic 396
- types de format
  - date 145, 227
  - devise 145, 227
  - texte 145, 227

## U

- UDA-SQL-0107 396
- UDA-SQL-0114 396
- URI
  - impossible d'accéder au service 395
- utilisateurs
  - ajout 260
  - packs 259
- utilisation
  - ensembles de données multiples 296
  - plusieurs connexions de source de données 40, 43
- utilisation de chaînes imbriquées
  - dans les expressions 248
  - dans les macros 248
- utilisation de chaînes littérales
  - dans les expressions 248
  - dans les macros 248
- utilisation de guillemets
  - dans les expressions 248
  - dans les macros 248

## V

- valeurs NULL
  - SAP BW 309

- validation
  - sujets de requête relationnelle 96
  - sujets de requête SAP BW 220
- vargraphic, type de données 396
- variables
  - instructions pour les données de faits SAP BW 656
  - procédure stockée 84
  - SAP BW 234
  - utilisation lors de l'externalisation 665
- variables SAP BW 211, 234
- vérification
  - modèles 256
  - modèles IBM Cognos ReportNet 383, 386
  - packs 256
  - projets 256
  - relations 68
- versions de modèle
  - mise à jour 270
- vue Avertissement de modèle 31
- vue d'importation 67
- vue métier 67
- vue métier pour métadonnées relationnelles 176
  - dossiers 185
  - espaces-noms 185
  - raccourcis 181, 183
  - raccourcis et relations 183
- vue métier pour métadonnées SAP BW 250
  - dossiers 252
  - espaces-noms 252
  - raccourcis 251
  - raccourcis et dimensions 251
- vues métier de modèle 176, 250