IBM Cognos Real-time Monitoring Workbench
Version 10.2.2

Guide de référence de la modélisation

IBM

Important
Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 471.

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.

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

Informations sur le produit

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.

Licensed Materials - Property of IBM. Eléments sous licence - Propriété d'IBM

© Copyright IBM Corporation 2007, 2014.

Table des matières

Avis aux lecteurs canadiens . . . xv

Introduction . . . xvii

Nouveautés . . . xix
Nouvelles fonctions de la version 10.2.2. . . . xix
Nouveautés de la version 10.2.1 . . . xix
Nouveautés de la version 10.2.0 . . . xix
Nouveautés de la version 10.1.1 . . . xix

Chapitre 1. Mise en route . . . 1
Lancement de Real-time Monitoring Workbench . . . 1
Utilisation du produit . . . 2

Chapitre 2. Filtres d'accès . . . 3
Conditions de filtre d'accès . . . 3
Dénomination des utilisateurs . . . 4
Table de correspondance dans les filtres d'accès. . . . 4
Tables de correspondance utilisateur dans les filtres d'accès. . . . 4
Fonctionnement et restrictions des filtres d'accès . . . 6
Création d'un filtre d'accès sur une vue . . . 7
Création de filtres d'accès basés sur des tables de correspondance . . . 7
Création d'un filtre d'accès simple . . . 8
Affectation d'un filtre d'accès aux utilisateurs et aux rôles . . . 8

Chapitre 3. Structure de l'adaptateur . . . 11
Création de la structure de l'adaptateur externe . . . 12
Plug-in de l'adaptateur externe . . . 12
Configuration du fichier de configuration . . . 14
Fichier ConfigXML . . . 15
Dossier des fichiers JAR . . . 16
Compilation et démarrage d'une application externe. . . . 16
Agents externes . . . 17
Flux de données externes. . . . 18
Tables de correspondance externes. . . . 20

Chapitre 4. Agents. . . . 25
Sources externes. . . . 25
Création d'agents . . . 28
Modification des agents . . . 29
Agents de fichier à plat . . . 30
Création d'un agent de fichier à plat . . . 30
Programme de l'agent de fichier . . . 31
Agent de système de fichiers . . . 34
Création d'un agent de système de fichiers . . . 34
Agents de file d'attente JMS (Java Messaging Service) . . . 35
Propriétés JNDI permettant la connexion à un espace-noms distant. . . . 36
Création d'un agent de file d'attente JMS . . . 37
Agents de rubrique JMS . . . 38
Création d'un agent de rubrique JMS. . . . 39
Agents TIBCO Rendezvous . . . 40
Attributs . . . 40
Création d'un agent TIBCO Rendezvous. . . . 41
Agents de services Web . . . 41

© Copyright IBM Corp. 2007, 2014

Attributs des agents de services Web . . . 42
Création d'un agent de service Web . . . 42

Chapitre 5. Alertes . . . 43
Création d'une alerte autonome. . . . 43
Création d'une alerte associée à une règle . . . 44
Clonage d'une alerte existante . . . 44
Attributs des alertes . . . 44
Objet et corps de texte d'un message . . . 45
Abonnés aux alertes . . . 46
Gestion des messages de notification d'alerte . . . 47
Etats des alertes . . . 47
Escalade d'alerte. . . . 48
Consolidation de plusieurs messages . . . 48
Configuration d'une alerte pour l'appel d'un service Web externe . . . 49
Fonction définie par l'utilisateur associée au service Web . . . 50
Ajout de tableaux récapitulatifs dans le corps de la notification d'alerte . . . 50
Filtrage des tableaux récapitulatifs. . . . 51
Option Données du tableau récapitulatif basées sur . . . 52

Chapitre 6. Règles métier . . . 55
A propos des règles métier . . . 55
Création d'une règle métier . . . 55

Chapitre 7. Infrastructure d'événements communs . . . 57
Fonctionnement de l'infrastructure d'événements communs . . . 57
Flux de données Infrastructure d'événements communs . . . 57
Prérequis à la création d'un flux de données CEI . . . 58
Création d'une table de flux de données à partir d'une connexion d'infrastructure d'événements communs . . . 60
Informations sur la colonne Infrastructure d'événements communs . . . 60
Intervalle de suppression de l'état . . . 61

Chapitre 8. Cubes . . . 63
Mesures . . . 64
Dimensions dans les cubes . . . 64
Création de cubes . . . 65
Colonnes Mesure . . . 66
Création d'un cube avec des colonnes de dimension. . . . 66

Chapitre 9. Types de données . . . 69
Conversion des types de données . . . 69
Conversion explicite . . . 69
Conversion implicite . . . 69
Ordre de priorité . . . 70
Types de données C-SQL pour les valeurs numériques . . . 71
Types de données tiers correspondant aux types de données numériques C-SQL . . . 71
Combinaison de types numériques . . . 72
Conversion de types numériques . . . 72
Conversion de types numériques en chaînes . . . 73
Précision décimale des résultats . . . 73
Conversion . . . 73
Multiplication . . . 73
Division . . . 74
Addition et soustraction . . . 74
Autres fonctions. . . . 74
Chaîne . . . 74
Largeur des chaînes . . . 75
Types de données tiers correspondant au type VARCHAR. . . . 75
Concaténation de chaîne . . . 75
Littéral de chaîne . . . 75

Conversion de chaînes dans d'autres types de données . . . 76
Date-heure . . . 76
Conversion des valeurs date-heure en chaînes . . . 76
Comparaison des valeurs date-heure . . . 77
Arithmétique appliquée aux valeurs date-heure . . . 77
Types de données tiers correspondant au type date-heure C-SQL . . . 77
Littéral TIMESTAMP . . . 78
Littéral INTERVAL . . . 78
Format date-heure . . . 81
Opérateurs booléens . . . 82

Chapitre 10. Flux de données . . . 85
Fonctionnement des flux de données . . . 85
Sources d'événements externes pour les flux de données . . . 86
Attributs de flux de données . . . 87
Création de tables de flux de données . . . 87
Modification des tables de flux de données . . . 89

Chapitre 11. Dimensions . . . 91
Hiérarchie de niveaux . . . 92
Noms d'alias dans les cubes . . . 92
Zone Trier par colonne . . . 92
Catégories géographiques . . . 93
Colonnes clés . . . 93
Création de dimensions . . . 94

Chapitre 12. Système de fichiers Cognos Real-time Monitoring . . . 97
Tables de correspondance établies à partir de fichiers à plat . . . 97
Onglet Informations sur la zone . . . 99
Onglet Mise en cache des données . . . 100
Création d'une table de correspondance à partir d'un fichier à plat . . . 101

Chapitre 13. Fichiers à plat . . . 103
Données de fichiers à plat . . . 103
Fonctionnement des fichiers à plat . . . 103
Tables de flux de données associées aux fichiers à plat . . . 104
Fichiers délimités . . . 110
Fichiers à largeur fixe . . . 110
Prise en charge des fichiers XML . . . 111
Emplacement XPath des colonnes et des lignes dans les fichiers XML . . . 112
Informations sur la zone XML . . . 112
Types de données XML . . . 113
Procédure de mise à plat des fichiers XML . . . 115

Chapitre 14. Formules . . . 125
Fonctions . . . 125
Types de fonction . . . 125
Fonctions scalaires . . . 125
Fonctions d'ensembles . . . 126
Fonctions de classement . . . 127
Fonctions d'ensembles mobiles . . . 127
Fonctions d'ensembles en défilement continu . . . 128
Catégories de fonctions . . . 128
Fonctions mathématiques . . . 129
Fonctions de classement . . . 129
Fonctions de règles . . . 130
Fonctions statistiques . . . 130
Fonctions Texte et Chaîne . . . 130
Fonctions Séries chronologiques et agrégation . . . 131
Fonctions des vues . . . 131

Chapitre 15. Fonctions C-SQL . . . 133
MOV_function . . . 136
Syntaxe des fonctions MOV_function . . . 137
Séries chronologiques. . . . 139
Avertissement concernant les vues . . . 140
Interactions GROUP BY . . . 140
Instruction CASE . . . 141
CASE . . . 141
ELSE . . . 142
END . . . 142
THEN . . . 142
WHEN . . . 142
Conversion . . . 142
CAST . . . 142
Date et heure . . . 144
CURRENT_TIMESTAMP . . . 144
DATE_ADD . . . 144
DATE_DIFF . . . 145
GREATEST . . . 146
LAST_DAY . . . 147
LEAST . . . 147
PRIOR_VALUE . . . 148
TIMESTAMP_DIFF . . . 149
TO_CHAR . . . 150
TO_DATE . . . 151
Nombres . . . 151
ABS . . . 152
CEIL . . . 152
DISPLAY_MONEY . . . 153
EXP . . . 155
FLOOR . . . 155
GREATEST . . . 156
LEAST . . . 156
LOG . . . 157
MOD . . . 158
POWER . . . 159
PRIOR_VALUE . . . 159
ROUND . . . 160
SAFE_DIVIDE . . . 161
SIGN . . . 162
SQRT . . . 163
TRUNC . . . 163
IS_ACKED . . . 164
IS_RAISED . . . 165
PRIOR_VALUE . . . 165
Système . . . 166
CURRENT_USER . . . 166
Texte . . . 167
CHAR_LENGTH . . . 167
CONCAT . . . 168
GREATEST . . . 168
LEAST . . . 169
LOWER . . . 170
LPAD . . . 170
LTRIM . . . 171
POSITION . . . 172
PRIOR_VALUE . . . 172
RPAD . . . 173
RTRIM . . . 174
SUBSTRING . . . 175
UPPER . . . 176

Séries chronologiques et agrégation . . . 177
AVG . . . 177
COUNT . . . 178
CURRENT . . . 179
MAX . . . 179
MIN . . . 180
MOV_AVG . . . 181
MOV_COUNT . . . 182
MOV_MAX . . . 183
MOV_MIN . . . 184
MOV_SUM . . . 186
PREV . . . 187
SUM . . . 188
TUMBLE_AVG . . . 189
TUMBLE_COUNT. . . . 190
TUMBLE_MAX. . . . 191
TUMBLE_MIN . . . 193
TUMBLE_SUM . . . 194
Séries chronologiques et statistiques . . . 196
MOV_STD_DEVIATION. . . . 196
MOV_VARIANCE . . . 197
NTILE. . . . 198
RANK. . . . 200
RATIO_TO_REPORT . . . 201
STD_DEVIATION . . . 202
SUM_OVER_GROUPS . . . 203
TUMBLE_STD_DEVIATION . . . 204
TUMBLE_VARIANCE . . . 206
VARIANCE . . . 207
Fonctions fournies comme exemples de fonctions UDF . . . 208
concatList . . . 208
concatSet . . . 209
gammaDist . . . 210
logNormDist . . . 210
median . . . 211
mode . . . 212
yield . . . 213

Chapitre 16. HTTP Post . . . 215
Fonctionnement de HTTP Post . . . 215
Tables de flux de données HTTP Post . . . 215
Création d'une table de flux de données HTTP Post . . . 217
Informations sur les colonnes HTTP Post . . . 217
Envoi vers une table de flux de données HTTP Post . . . 218
Envoi vers des zones message . . . 219
Envoi de valeurs dans l'URL . . . 220
Evénements d'entrée multiples . . . 220

Chapitre 17. Java Messaging Service (JMS) . . . 221
Flux de données JMS . . . 221
Création d'une table de flux de données à partir d'une connexion JMS (Java Messaging Service) . . . 223
Informations sur la colonne dans Java Messaging Service. . . . 224
Types de données JMS (Java Messaging Service). . . . 224

Chapitre 18. Connectivité JDBC (Java Database Connectivity) . . . 227
Tables JDBC (Java Database Connectivity) . . . 227
Tables de correspondance . . . 227
Création d'une source JDBC (Java Database Connectivity) pour une table de flux de données . . . 228
Création d'une source JDBC (Java Database Connectivity) pour une table de correspondance. . . . 229
Tables de flux de données . . . 229

Source de requête . . . 232
Exemple de table de flux de données utilisant une requête d'interrogation . . . 233
Interrogation de la source JDBC (Java Database Connectivity) . . . 233
Upsert . . . 234
Source de procédure stockée . . . 235
Exemple de réception de données de table de correspondance à l'aide d'une procédure stockée . . . 236
Exemple de réception d'un événement à l'aide d'une procédure stockée . . . 237
Mappage des types de données JDBC (Java Database Connectivity) . . . 237
Agents JDBC (Java Database Connectivity) . . . 238
Création d'un agent JDBC . . . 239

Chapitre 19. Accès JDBC (Java Database Connectivity) aux données des vues et des cubes . . . 243
Chemin d'accès aux classes . . . 243
URL JDBC (Java Database Connectivity) . . . 243
Interfaces des vues JDBC (Java Database Connectivity) . . . 244
Mappages de types de données . . . 245
Récapitulatif des colonnes de la fonction getColumns() . . . 245
Récapitulatif des colonnes de la fonction getTables() . . . 246
Valeurs renvoyées par la colonne DATA_TYPE . . . 247
Exemples d'accesseurs JDBC . . . 248
Java . . . 248
Exemple : Etablissement d'une connexion aux serveurs IBM Cognos Real-time Monitoring . . . 248
Exemple : Interrogation du contenu d'une vue . . . 249
Exemple : Interrogation des spécifications des colonnes d'une vue . . . 251
Exemple : Interrogation des métadonnées des colonnes . . . 252
Exemple : Interrogation des métadonnées des vues . . . 253
Interfaces de cube JDBC (Java Database Connectivity) . . . 255
IOlapStatement . . . 255
IOlapDatabaseMetadata . . . 255
Exemple : Interrogation de tous les cubes . . . 256
Exemple : Interrogation des métadonnées d'un cube . . . 257
Exemple : Récupération des membres enfant d'un niveau dans un cube . . . 260
Exemple : Interrogation des données de cube à l'aide du format de requête XML . . . 261
Exemple : Récupération des données d'un cube à l'aide de requêtes MDX . . . 262

Chapitre 20. Tables de correspondance . . . 265
Fonctionnement des tables de correspondance . . . 265
Création de tables de correspondance . . . 266
Modification des tables de correspondance . . . 267
Limitations de la colonne dans les requêtes de tables de correspondance . . . 268
Mise en cache des requêtes de tables de correspondance . . . 269
Mise en cache sur demande . . . 270
Mise en cache prefetch . . . 270
Mise en cache dans la bibliothèque JDBC (Java Database Connectivity) . . . 271

Chapitre 21. Objets dans Cognos Real-time Monitoring Workbench . . . 273
Etat d'un objet . . . 273
Noms d'objet . . . 274
Termes réservés . . . 274
Espace-noms d'objets . . . 281
Exportation sélective d'objets . . . 282
Exemple : Exportation d'objets avec des dépendances . . . 283
Exemple : Exportation d'objets avec des exigences . . . 285
Exemple : Exportation d'objets avec des exigences et des dépendances . . . 286
Exportation d'objets sélectionnés . . . 288

Chapitre 22. Opérateurs et constantes . . . 291
Opérateurs numériques . . . 291
Opérateurs de chaîne . . . 291

Opérateurs de comparaison . . . 292
Opérateur LIKE . . . 292
Opérateurs logiques . . . 293
Constantes . . . 294

Chapitre 23. Autorisations . . . 295
Droits d'accès . . . 295
Droits d'accès au niveau du type . . . 296
Droits d'accès à certains objets . . . 297
Autorisation de création d'objets . . . 299
Affichage des types d'objets que vous êtes autorisé à créer . . . 299
Droits d'autorisation . . . 299
Restrictions d'autorisation . . . 300
Héritage et dépendances des autorisations . . . 300
Dépendances . . . 300

Chapitre 24. Intégration du serveur de portail . . . 303
Prérequis pour l'intégration du serveur de portail . . . 303
Intégration dans le serveur de portail BEA WebLogic . . . 303
Conversion du fichier lavaJSR168.war . . . 303
Déploiement du fichier lavaJSR168.war dans le serveur de portail WebLogic . . . 304
Intégration dans le serveur de portail IBM WebSphere . . . 304
Intégration dans le serveur de portail JBoss . . . 305
Instances de portlet multiples sur un serveur JBoss Portal . . . 305

Chapitre 25. Processus . . . 307
Fonctionnement des processus . . . 307
Création et utilisation des processus . . . 308
Tables de flux de données . . . 309
Recherche de table de correspondance . . . 309
Fichier de définition de processus . . . 309
Création de définitions de processus . . . 311
Création de diagrammes de processus . . . 311

Chapitre 26. Tableaux récapitulatifs . . . 313
Création de tableaux récapitulatifs . . . 313
Attributs des tableaux récapitulatifs . . . 314
Attributs des tableaux récapitulatifs externes . . . 314
Vues de tableau récapitulatif . . . 316

Chapitre 27. Rôles . . . 317
Création de rôles . . . 317
Attributs de rôle . . . 318
Onglet Droits d'accès . . . 318

Chapitre 28. Règles . . . 321
Création de règles . . . 321
Clonage de règles . . . 322
Attributs de règle . . . 322
Conditions de règle . . . 323
Actions de règle . . . 324
Exemples de règle . . . 324
Règles contrôlant les alertes . . . 325
Alertes spécifiques . . . 325
Journal système . . . 326

Chapitre 29. Salesforce . . . 327
Création d'un agent Salesforce . . . 327
Importation de tables de correspondance et de flux de données Salesforce . . . 327

Fonction de mise à plat Salesforce . . . 328
Fonction de liste de sélection Salesforce . . . 330
Console d'administration Salesforce . . . 331

Chapitre 30. Connectivité SAP . . . 333
Création d'un agent SAP . . . 333
Importation d'objets ODS (Operational Data Store) . . . 334
Importation de cubes OLAP . . . 334

Chapitre 31. Scénarios . . . 335
Création de scénarios . . . 336

Chapitre 32. Instructions SELECT . . . 337
Syntaxe . . . 337
Liste Select . . . 338
Expression CASE . . . 339
Clause FROM . . . 340
Contraintes sur les vues . . . 340
Opérations de jointure . . . 340
Jointures internes . . . 341
Jointures externes . . . 341
Jointures imbriquées . . . 342
Expressions de table . . . 342
Syntaxe . . . 342
Restrictions . . . 342
Exemple de clause HAVING . . . 343
Clause WHERE. . . . 343
Prédicats . . . 344
Alias . . . 344
Clause GROUP BY . . . 344
Alias . . . 345
Vues dérivées . . . 345
Sémantique des vues avec état . . . 346
Clause ORDER BY . . . 346

Chapitre 33. TIBCO Rendezvous . . . 349
Fonctionnement de TIBCO Rendezvous . . . 349
Tables TIBCO Rendezvous . . . 349
Limitations . . . 349
Création d'une table de flux de données TIBCO Rendezvous . . . 350
Informations sur les colonnes TIBCO . . . 352
Types de données TIBCO Rendezvous . . . 354

Chapitre 34. Utilisateurs . . . 355
Modification de vos propres informations de compte . . . 355
Création, suppression ou modification d'un compte utilisateur . . . 355
Onglet Détails de l'utilisateur . . . 355
Onglet Profils de diffusion . . . 356
Courrier électronique . . . 356
Service Web . . . 356
Onglet Droits d'accès . . . 357

Chapitre 35. Fonctions définies par l'utilisateur . . . 359
Restrictions applicables aux fonctions définies par l'utilisateur (fonctions UDF) . . . 359
Création et utilisation d'une fonction définie par l'utilisateur . . . 360
Modification d'une fonction définie par l'utilisateur . . . 360
Fichiers manifestes . . . 361

Chapitre 36. Vues. 363
Création de vues 363
Copie d'une vue 364
Attributs de vue 365
Contraintes sur les vues 366
Jointures synchronisées 366
Restriction 367
Exemple de jointure synchronisée 367
Flux de données consolidés. 369
Vues agrégées 369
Mise à jour des vues par le biais de la propagation d'événements 370
Vues sans état et vues avec état 370
Initialisation des vues 371
Conservation des événements dans des vues sans état. 371
Conservation des vues dans une base de données 372
Attributs de sauvegarde des vues 373
Colonnes des vues à sauvegarder. 374
Activation du changement de niveau pour l'accès au détail 374
Données IBM Cognos Real-time Monitoring dans IBM Cognos Business Intelligence 375
Affichage des données IBM Cognos Real-time Monitoring dans IBM Cognos Business Intelligence 375

Chapitre 37. Services Web 377
Flux de données de service Web 377
Attributs des flux de données pour les services Web 378
Attributs des flux de données pour les services Web d'interrogation 379
Création d'un flux de données de service Web 380
Tables de correspondance pour les services Web. 381
Création d'une table de correspondance de services Web 383
Colonnes de sortie. 384
Colonnes d'entrée 384
Modèles de substitution de la chaîne 385
Processus externes des services Web. 386
Publication d'un élément vers un service Web externe 386
Attributs de processus externe. 386
Création d'un processus externe 386
Implémentation du service externe 387
Zones de message 387
Fichier WSDL de service Web 388

Chapitre 38. Fenêtres de requêtes 391
Présentation des fenêtres de requêtes 391
Types de fenêtres 392
Déclarations et références des fenêtres 392
En ligne 392
Référence par nom 392
Plusieurs fenêtres par requête 393
Extension d'une définition de fenêtre avec une autre 393
Restrictions 393
Fenêtres de série d'événements 394
Clause EVENTS 394
Evénement en cours 395
Fenêtres chronologiques 396
Clause RANGE. 396
Quels sont les événements inclus ? 398
Clause ORDER BY 399
Arrivée dans n'importe quel ordre 399
Ordre décroissant 400
Horodatages de valeur NULL 400
Séries chronologiques de nombres entiers 401
Partitions des fenêtres 401

Clause PARTITION BY . . . 402
Mise à jour des vues pour une clause GROUP BY simple . . . 402
Mise à jour des vues pour une partition avec référence de cadre . . . 402
Mise à jour des vues pour une partition avec référence d'opérateur . . . 403
Avantage des partitions sur les groupes . . . 404
Utilisation de la clause GROUP BY dans les fenêtres sur le point d'expirer . . . 404
Résultats historiques des vues partitionnées . . . 404
Avancement des fenêtres . . . 405
Clause SLIDE . . . 405
Fenêtres en défilement continu . . . 406
Fenêtres Défilement continu de fin . . . 407
Fonctions tumble . . . 407
Référence de mise à jour de fenêtre . . . 408
Clause REFERENCE . . . 408
Initialisation de fenêtre . . . 409
Clause INITIALIZE . . . 409
Exemple de clause INITIALIZE . . . 410

Chapitre 39. Fichiers XML et XSD utilisés par IBM Cognos Real-time Monitoring . . . 411
Fichiers XML et XSD dans Cognos Real-time Monitoring . . . 411
Dépendances . . . 411
Espace . . . 411
Caractères d'échappement . . . 412
Données de type caractères . . . 412
Téléchargement de fichiers XML . . . 412
Téléchargement de fichiers XML à partir d'une ligne de commande . . . 413
Téléchargement de fichiers XML à partir d'un navigateur Web . . . 413
Définition d'un objet avec le langage XML . . . 414
Exemple : Création d'un utilisateur . . . 414
Définition de plusieurs objets avec le langage XML . . . 415
Exemple : Commande par lots . . . 415
Modification d'un objet existant avec le langage XML . . . 416
Dépendances . . . 417
Emission de commandes XML . . . 418
Exemple : Activation d'un objet et de ses dépendances . . . 418
Fichiers XSD Cognos Real-time Monitoring . . . 418

Annexe A. Méthodes JDBC (Java Database Connectivity) . . . 423
Driver . . . 423
Connection . . . 423
Connexion . . . 424
Statement . . . 427
DatabaseMetaData . . . 430
ResultSet . . . 443
ResultSetMetaData . . . 452

Annexe B. API UM-SSO . . . 455
Fonctionnement du plug-in UM-SSO . . . 455
Connexion initiale à IBM Cognos Real-time Monitoring . . . 455
Connexion à IBM Cognos Real-time Monitoring et accès à l'application externe . . . 455
Connexion à l'application externe et accès à IBM Cognos Real-time Monitoring . . . 456
Synchronisation des utilisateurs . . . 456
Installation d'un plug-in UM-SSO . . . 456
Fichier XML setPropertyDetails . . . 457
Eléments du fichier XML setPropertyDetails . . . 457
UMAPI . . . 460
doAuthenticate . . . 460
doLogout . . . 461
fetchAllUsers . . . 462
fetchRoles . . . 462

getResourceBundle . . . 463
getRoles . . . 463
getSettings . . . 464
onPropertyChange . . . 464
Données d'identification . . . 464
getPassword . . . 464
getUserName . . . 465
setPassword . . . 465
setUserName . . . 465
Rôle . . . 465
getAttributes . . . 465
getRoleDescription . . . 466
getRoleName . . . 466
getUsers . . . 466
setAttributes . . . 466
setRoleDescription . . . 466
setRoleName . . . 467
setUsers . . . 467
SessionInfo . . . 467
getAttributes . . . 467
getUser . . . 467
setAttributes . . . 468
setUser . . . 468
Utilisateur . . . 468
getAttributes . . . 468
getUserDescription . . . 468
getUserId . . . 469
setAttributes . . . 469
setUserDescription . . . 469
setUserId . . . 469
Exceptions . . . 470
ApplicationNotAvailableException . . . 470
InvalidCredentialException . . . 470
SessionTimedOutException . . . 470

Remarques . . . 471

Glossaire . . . 475
A . . . 475
C . . . 475
D . . . 475
E . . . 475
F . . . 475
J . . . 476
M . . . 476
N . . . 476
P . . . 476
S . . . 476
T . . . 476
U . . . 476
V . . . 476

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.

© Copyright IBM Corp. 2007, 2014

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

Le présent document est destiné à être utilisé avec IBM® Cognos Real-time Monitoring Workbench.

Public concerné

Ce document s'adresse aux administrateurs et aux modélisateurs.

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).

Fonctions d'accessibilité

IBM Cognos Real-time Monitoring Workbench ne prend pas en charge les fonctions d'accessibilité permettant aux utilisateurs souffrant d'un handicap physique, telle une mobilité réduite ou une vision limitée, d'utiliser le produit.

Déclarations préliminaires

La présente documentation décrit les fonctionnalités actuelles du produit. Des références à des éléments actuellement non disponibles peuvent être incluses. Aucune implication de disponibilité future ne doit en être déduite. Aucune de ces références n'est constitutive d'un engagement, d'une promesse ou d'une obligation légale de fournir des matériels, codes ou fonctionnalités d'aucune sorte. Le développement, la diffusion et la planification des caractéristiques ou fonctionnalités restent à la totale discrétion d'IBM.

© Copyright IBM Corp. 2007, 2014

Nouveautés

Cette section contient une liste des nouvelles fonctions, de celles changées et supprimées pour cette nouvelle version. Vous pourrez ainsi plus facilement planifier vos mises à niveau et vos stratégies de déploiement d'application, ainsi que les critères de formation requis pour vos utilisateurs.

Pour plus d'informations sur la mise à niveau, voir le *Guide d'installation et de configuration* d'IBM Cognos Business Intelligence.

Pour obtenir des informations sur les autres nouveautés de cette édition, voir le *Guide des nouveautés* d'IBM Cognos Business Intelligence.

Les informations relatives aux nouveautés des versions précédentes sont disponibles dans la documentation à laquelle vous pouvez accéder via le site Web IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter).

Pour consulter la liste à jour des environnements pris en charge par les produits IBM Cognos, tels que les systèmes d'exploitations, les correctifs, les navigateurs, les serveurs Web, les serveurs d'annuaire, les serveurs de base de données et les serveurs d'application, consultez le site IBM Cognos Customer Center (http://www.ibm.com/software/data/cognos/customercenter).

Nouvelles fonctions de la version 10.2.2

Cette édition d'IBM Cognos Real-time Monitoring Workbench ne comporte aucune nouvelle fonction.

Nouveautés de la version 10.2.1

Cette édition d'IBM Cognos Real-time Monitoring Workbench ne comporte aucune nouvelle fonction.

Nouveautés de la version 10.2.0

La liste ci-dessous répertorie toutes les nouveautés ajoutées depuis la dernière version.

Workbench - Guide de référence technique et guide de référence de la modélisation

L'ancien document IBM Cognos Real-time Monitoring Workbench Technical Reference est désormais disponible en français sous le nom IBM Cognos Real-time Monitoring Workbench - Guide de référence de la modélisation.

Nouveautés de la version 10.1.1

La liste ci-dessous répertorie les nouveautés de la version 10.1.1.

© Copyright IBM Corp. 2007, 2014

Intégration renforcée entre IBM Cognos Real-time Monitoring Dashboard et IBM Cognos Business Insight

Lorsque vous utilisez des objets du tableau de bord IBM Cognos Real-time Monitoring dans IBM Cognos Business Insight, vous pouvez :

- Passer d'un type d'affichage à un autre
- Modifier le style des mesures d'un affichage
- Modifier des propriétés spécifiques d'un affichage
- Utiliser les widgets de filtre d'IBM Cognos Business Insight

Pour plus d'informations, voir les rubriques relatives à la modification des type d'affichage, des styles de mesure et des propriétés spécifiques aux graphiques, ainsi qu'aux widgets de filtrage, dans le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Dashboard*.

Les tables de correspondance de flux de données prennent en charge les dimensions en ligne

Grâce à l'utilisation d'une table de correspondance de flux de données basée sur un flux d'événements, les valeurs de la table sont mises à jour dès que les données sont traitées, et peuvent être extraites de sources non-JDBC, telles que JMS ou WebSphere MQ. Vous pouvez désormais utiliser des dimensions en ligne dans les cubes et dans les vues, sans qu'il soit nécessaire de créer des jointures avec la dimension - vous pouvez créer la dimension dans une table de correspondance en suivant la même procédure que pour une dimension classique. Pour plus d'informations, voir la rubrique relative à la création des tables de correspondance dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Prise en charge de la fonction upsert étendue aux fichiers à plat et aux formats de service de messagerie JDBC

Dans la version 10.1.0, la fonctionnalité upsert, qui combine les actions de mise à jour, d'insertion et de suppression de données provenant des tables de flux de données, n'était disponible que pour les flux de données JDBC (Java™ Database Connectivity). La version 10.1.1 étend cette fonctionnalité aux flux de fichiers à plat et aux flux JMS, entre autres. Pour plus d'informations, voir les rubriques relatives à l'activation de la fonction upsert pour un flux de données dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*. Consultez également la rubrique dédiée à la fonction upsert dans le manuel *IBM Cognos Real-time Monitoring Workbench - Référence de modélisation*.

Recherche d'objets

Dans les éditions précédentes, il pouvait être fastidieux de rechercher des objets particuliers dans des listes triées. Vous pouvez désormais rechercher du texte spécifique contenu dans la définition XML. Pour plus d'informations, voir la rubrique relative aux recherches dans la bibliothèque d'objets dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Affichage des dépendances et des exigences des objets

Vous pouvez utiliser l'onglet **Relations** pour visualiser l'ensemble des dépendances et des exigences d'un objet. Pour plus d'informations, voir la rubrique relative aux dépendances et aux exigences des objets dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Conception de la structure d'un adaptateur externe

Vous pouvez configurer un adaptateur externe de manière à convertir des données d'un format qui n'était pas pris en charge auparavant (par exemple, JSON ou RSS Feed) vers un format pris en charge par RTM (par exemple, tabulaire, fichier à plat, XML). Pour plus d'informations, voir la rubrique relative aux structures d'adaptateur dans le manuel *IBM Cognos Real-time Monitoring Workbench - Référence de modélisation*.

Chapitre 1. Mise en route

Votre administrateur système doit au préalable installer, configurer et démarrer les composants serveur IBM Cognos Real-time Monitoring. Avant de recevoir des événements ou de récupérer des données de tables de correspondance, les serveurs doivent localiser et collecter les données. Par exemple, pour accéder à une base de données en vue de récupérer des données de tables de correspondance, les serveurs doivent être configurés avec le nom et le mot de passe d'accès à la base de données.

Pour plus d'informations, voir le *Guide d'installation et de configuration* d'IBM Cognos Real-time Monitoring.

Lancement de Real-time Monitoring Workbench

Si vous êtes un administrateur système, commencez par accéder à IBM Cognos Real-time Monitoring Workbench pour pouvoir configurer les paramètres système avant que d'autres utilisateurs n'accèdent au produit.

Avant de commencer

Avant de lancer IBM Cognos Real-time Monitoring Workbench, assurez-vous que les conditions suivantes sont remplies :

- Votre navigateur Web est Microsoft Internet Explorer version 6.0 ou ultérieure.
- Les serveurs Cognos Real-time Monitoring sont en cours d'exécution.
- JavaScript est activé.
- Adobe Flash version 9 ou suivante est installé.

Pourquoi et quand exécuter cette tâche

Pour vous connecter, utilisez le compte utilisateur **rtmadmin** et le mot de passe **manager**. Pour plus d'informations, voir la section relative à la configuration de Cognos Real-time Monitoring après installation, dans le *Guide d'installation et de configuration d'IBM Cognos Real-time Monitoring*.

Lorsque le mode sécurisé est activé, Cognos Real-time Monitoring Workbench affiche une bannière indiquant que le système est en mode sécurisé et que tous les événements sont suspendus. Le flux de données s'interrompt lorsque le système est en mode sécurisé. Pour plus d'informations, voir le *Guide d'installation et de configuration* d'IBM Cognos Real-time Monitoring.

Procédure

1. Dans votre navigateur Internet, saisissez l'URL suivante, où *nom_serveur* désigne l'ordinateur hôte et le numéro du port :

   http://*nom_serveur*/cognos/realtime/landing/landingpage.htm

   La page d'accueil de Cognos Real-time Monitoring s'ouvre.
2. En fonction de votre rôle (administrateur, modélisateur ou développeur), cliquez sur l'un des liens suivants :
   - **Gérer le contenu en temps réel**
   - **Modéliser mes données en temps réel**

© Copyright IBM Corp. 2007, 2014

- **Créer mon tableau de bord en temps réel**

3. Connectez-vous à l'aide du nom d'utilisateur et du mot de passe attribués par votre administrateur.

   Remarque : Si le code d'accès unique entre IBM Cognos Business Intelligence et Cognos Real-time Monitoring est activé et que vous êtes déjà connecté à IBM Cognos Business Intelligence, la page de connexion ne s'affiche pas. Vous êtes automatiquement connecté à Cognos Real-time Monitoring Workbench.

Utilisation du produit

Vous pouvez exécuter les tâches suivantes à l'aide d'IBM Cognos Real-time Monitoring Workbench.

- Configuration de comptes utilisateur.

  Chaque utilisateur de Cognos Real-time Monitoring Workbench ou de Cognos Real-time Monitoring Dashboard doit posséder un compte utilisateur. Les utilisateurs peuvent consulter les informations relatives à leur compte en cliquant sur **Paramètres de compte** dans l'application.

  Les administrateurs système peuvent ajouter et modifier les comptes utilisateur à l'aide de la **Console d'administration** .

- Création et modification d'agents.

  Les agents collectent les événements et les données de la table de correspondance pour les rendre disponibles dans Real-time Monitoring.

  Les administrateurs système peuvent créer et modifier les agents à l'aide de la **Console d'administration**.

- Création de flux de données et de tables de correspondance.

  Les agents insèrent les données dans des tables de flux de données et des tables de correspondance. Ces tables représentent la source des vues métier où les informations sont présentées et agrégées.

  Vous pouvez créer des flux de données et des tables de correspondance à l'aide de **Workbench**.

- Utilisation de vues métier.

  Les vues métier contiennent, agrègent et fournissent des informations sur les flux de données et les tables de correspondance.

  Vous pouvez créer des vues métier complexes grâce au **Modélisateur de scénarios**.

- Création de scénarios, de règles et d'alertes.

  Vous pouvez utiliser Cognos Real-time Monitoring pour identifier les événements métier exceptionnels et tenir les utilisateurs au courant de l'activité.

  Pour cela, créez des scénarios, des règles et des alertes dans **Modélisateur de scénarios**. Vous pouvez également utiliser le **Modélisateur de scénarios** pour définir les tableaux récapitulatifs qui fournissent les mesures associées aux événements.

Chapitre 2. Filtres d'accès

Les filtres d'accès permettent à différents utilisateurs de visualiser plusieurs lignes de la même vue ou du même cube en fonction des critères indiqués dans le filtre. Ces filtres limitent les données visibles par un utilisateur sans qu'il soit nécessaire de définir une nouvelle vue ou un nouveau cube pour chaque utilisateur.

Prenons l'exemple ci-après d'une vue du total des ventes par région :

```
Total Sales Region
----------- -------
  763000.00 West
  489500.00 Central
  522950.00 South
  650740.00 East
```

La définition d'un filtre d'accès tel que `OrderTotals.Region=Employees.Region` vous permet de limiter la visibilité des utilisateurs aux lignes qui s'appliquent à leur région d'activité. En conséquence, un utilisateur de la région Central obtiendrait le résultat suivant :

```
Total Sales Region
----------- -------
  489500.00 Central
```

Les filtres constituent des expressions logiques semblables à la clause WHERE d'une définition de vue. Pour obtenir une description complète des filtres, voir «Conditions de filtre d'accès». Pour plus d'informations sur les clauses WHERE, voir «Clause WHERE», à la page 343.

Les filtres d'accès se définissent sur chaque vue et chaque cube individuellement et s'appliquent aux utilisateurs et aux rôles disposant d'un accès **Filtré/Lecture seule** à la vue ou au cube. Pour en savoir davantage sur le fonctionnement des filtres d'accès, voir «Fonctionnement et restrictions des filtres d'accès», à la page 6.

Pour appliquer un filtre d'accès à une vue ou à un cube, vous devez commencer par créer un filtre, puis l'affecter à des utilisateurs ou à des rôles en suivant les procédures décrites aux sections ci-après :

- «Création d'un filtre d'accès sur une vue», à la page 7
- «Affectation d'un filtre d'accès aux utilisateurs et aux rôles», à la page 8

Conditions de filtre d'accès

Les conditions de filtre d'accès sont des expressions logiques qui s'appliquent à chacune des lignes d'une vue ou à chacun des niveaux de dimension d'un cube.

Un utilisateur examinant la vue ou le cube ne voit que les lignes pour lesquelles l'expression a été évaluée comme étant vraie.

Chaque filtre doit contenir au minimum une condition évaluant les données présentes dans la vue ou dans le cube. Par exemple, la condition simple ci-après n'affiche que les lignes de la vue OrderTotals applicables à la région d'activité East : **`OrderTotals.Region='East'`**

© Copyright IBM Corp. 2007, 2014

Les filtres d'accès peuvent inclure des opérateurs booléens (AND, OR et NOT) et peuvent utiliser des parenthèses pour les regroupements. Par exemple, OrderTotals.Region='East' AND (CURRENT_USER()='Skyler' OR CURRENT_USER()='Nina').

Dénomination des utilisateurs

Un filtre d'accès peut inclure les noms d'utilisateurs spécifiques.

Un filtre d'accès, tel que `OrderTotal.Region='East'`, doit être affecté à chaque utilisateur ou rôle dans la région East (Est) afin de limiter l'accès à cet utilisateur. Il existe une expression plus puissante qui désigne les utilisateurs. La fonction CURRENT_USER (voir «CURRENT_USER», à la page 166) renvoie le nom de connexion de l'utilisateur examinant la vue. Vous pouvez inclure cette fonction dans la condition de filtre pour appliquer le filtre à des utilisateurs spécifiques.

Par exemple, la condition ci-après identifie également deux utilisateurs. Ces utilisateurs seront donc les seuls à pouvoir visualiser les résultats de la région East :

```
OrderTotals.Region='East' AND
(CURRENT_USER()='Skyler' OR CURRENT_USER()='Nina')
```

Les deux exemples ci-dessus sont limités par le fait que leurs expressions comportent des valeurs littérales codées en dur : le nom de la région et le nom des utilisateurs. L'utilisation de valeurs littérales se révèle problématique car vous devez modifier les filtres chaque fois que les noms changent.

Il existe une expression plus puissante qui peut être appliquée à tous les utilisateurs en récupérant dynamiquement les informations relatives aux utilisateurs et en les appliquant à la vue.

Table de correspondance dans les filtres d'accès

Outre les données de la vue active, les filtres d'accès peuvent récupérer les données d'une table de correspondance.

Si vous définissez une table de correspondance contenant des informations sur les utilisateurs, vous pouvez comparer ces dernières aux données de la vue afin de créer un filtre de table de correspondance dynamique.

Prenons l'exemple du filtre ci-après utilisant une table de correspondance Employees :

```
OrderTotals.Region=Employees.Region AND
CURRENT_USER()=Employees.User_Name
```

Si vous choisissez d'appliquer ce filtre à un grand nombre d'utilisateurs et de rôles, seuls les utilisateurs affectés à la même région d'activité que les données pourront visualiser ces dernières. Vous trouverez des exemples dans les sections suivantes.

Pour utiliser des recherches dynamiques, vous devez présenter les informations dans une table de correspondance externe.

Tables de correspondance utilisateur dans les filtres d'accès

En général, les tables de correspondance prennent en charge des tables de flux de données en fournissant des informations complémentaires sur ces dernières.

Lorsque vous l'utilisez dans un filtre d'accès, une table de correspondance contient des informations compatibles avec le filtre, c'est-à-dire, des informations sur l'utilisateur en cours.

En conséquence, une table de correspondance d'utilisateurs doit comporter au moins une colonne contenant un nom d'utilisateur identique au nom dont se sert l'utilisateur pour se connecter à IBM Cognos Real-time Monitoring.

Notez que CURRENT_USER() renvoie le nom de connexion de l'utilisateur tel qu'il est défini dans IBM Cognos Real-time Monitoring, en respectant la distinction majuscules-minuscules, et tel qu'il apparaît dans Cognos Real-time Monitoring Workbench. La distinction majuscules-minuscules doit donc être scrupuleusement respectée. Certains SGBD effectuant des comparaisons sans tenir compte de cette distinction, cette mise en garde peut ne pas s'appliquer à votre installation. Pour éviter tout risque d'absence de concordance, vous pouvez choisir de saisir les noms uniquement en minuscules ou en majuscules dans la table de correspondance, puis d'utiliser UPPER() ou LOWER() dans l'expression du filtre, comme dans l'exemple ci-après :

```
UPPER(CURRENT_USER())=Employees.User_Name
```

Vous ne pouvez pas utiliser UPPER() ou LOWER() dans la référence à la table de correspondance Employees. Pour plus de détails, voir «Limitations de la colonne dans les requêtes de tables de correspondance», à la page 268.

De la même façon, toutes les colonnes de texte référencées dans un filtre doivent faire l'objet d'une distinction majuscules-minuscules.

Pour utiliser une table de correspondance dans un filtre d'accès, ajoutez la table à l'ensemble de tâches du filtre avant de définir le filtre. Pour plus de détails, voir «Création d'un filtre d'accès sur une vue», à la page 7. Pour plus d'informations sur les tables de correspondance, voir Chapitre 20, «Tables de correspondance», à la page 265.

Une table de correspondance d'utilisateurs doit inclure les conditions suivantes :

- Une ligne pour chaque utilisateur doté d'un accès Filtré/Lecture seule. Si un utilisateur ne figure pas dans la table de correspondance, le filtre ne trouve aucune ligne concernant cet utilisateur.

  Pour plus d'informations sur les autorisations, voir «Droits d'accès», à la page 295.
- Au moins une colonne de texte contenant le nom de connexion de l'utilisateur. Si le SGBD fait la distinction entre les majuscules et les minuscules, entrez les noms tels qu'ils sont définis dans Cognos Real-time Monitoring, ou bien uniquement en minuscules ou en majuscules si vous prévoyez d'utiliser UPPER() ou LOWER() dans vos filtres.
- Une colonne pour chaque référence apparaissant dans le filtre. Les types de données doivent correspondre. En ce qui concerne les valeurs de caractère, les chaînes de la vue doivent être exactement identiques à celles de la table de correspondance.

Notez également qu'en cas de mise en cache des données de la table de correspondance, les filtres risquent de ne pas fonctionner si les données des utilisateurs ne figurent dans le cache. En d'autres termes, si vous ajoutez un utilisateur à la base de données, invalidez le cache de la table de correspondance

avant que l'utilisateur n'examine des vues ou des cubes filtré(e)s. Pour plus d'informations sur la mise en cache, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

Fonctionnement et restrictions des filtres d'accès

Les filtres d'accès ne s'appliquent que lorsqu'un utilisateur disposant d'un accès Filtré / Lecture seule sur une vue examine ou requiert des données de cette dernière, ou définit une nouvelle vue au-dessus de cette vue.

Les filtres n'ont aucune incidence sur les utilisateurs ou rôles dotés d'une autorisation en Lecture seule ou en Lecture et écriture sur cette vue et ne s'appliquent pas non plus aux utilisateurs recevant des tableaux récapitulatifs envoyés sous forme de pièces jointes dans des notifications d'alerte.

Lorsque vous faites appel aux filtres d'accès, rappelez-vous les points suivants :

- Par défaut, tous les nouveaux utilisateurs reçoivent l'autorisation d'accès "Aucun accès pour les nouveaux utilisateurs" pour les classes des objets Vue et Cube.

  Pour qu'un utilisateur puisse visualiser les résultats, il doit disposer, soit directement, soit en tant que membre d'un rôle, d'une autorisation en lecture seule ou en lecture et écriture pour les classes des objets Vue et Cube (totalité des vues et des cubes), et d'au moins un accès Filtré/Lecture seule sur la vue concernée.
- Lorsqu'un utilisateur fait l'objet de plusieurs filtres d'accès portant sur la même vue ou sur le même cube, par exemple en tant que membre de plusieurs rôles associés à différents filtres, il voit les lignes pour lesquelles l'un de ces filtres renvoie la valeur True.

  Par exemple, il est possible qu'un filtre n'autorise un utilisateur qu'à visualiser les données de la région West, mais qu'un autre filtre permette à cet utilisateur de visualiser tous les résultats relatifs à une gamme de produits spécifique. Dans ce cas, l'utilisateur verra tous les résultats associés à cette gamme de produits, quelle que soit la région.
- Les tableaux récapitulatifs contiennent toujours toutes les données de la vue à laquelle ils font référence, quels que soient les filtres d'accès associés à cette vue.

  Les utilisateurs qui reçoivent des tableaux récapitulatifs dans des notifications d'alerte visualisent toujours la totalité de la vue référencée par ces tableaux.
- Lorsqu'un utilisateur disposant d'un accès filtré à une vue crée une nouvelle vue par-dessus la vue filtrée, la nouvelle vue hérite des résultats filtrés de cet utilisateur, mais non de la définition du filtre.

  En conséquence, tout utilisateur examinant la vue dérivée verra les mêmes résultats filtrés que l'utilisateur ayant créé la vue. Par exemple, si l'utilisateur Skyler ne peut voir que la région West de la vue Total Sales, puis qu'il crée une nouvelle vue intitulée WrapUp, dérivée de la vue Total Sales, tout utilisateur autorisé à examiner la vue WrapUp verra les données de la région West, quel que soit l'accès dont il dispose sur la vue Total Sales.
- De la même façon, lorsqu'un utilisateur disposant d'un accès filtré à une vue crée une règle reposant sur cette vue, la règle hérite des résultats filtrés de cet utilisateur.

  En conséquence, cette règle ne voit que les événements qui remplissent la condition de filtre d'accès des utilisateurs, et tous les abonnés à l'alerte associée à la vue ne reçoivent que les alertes relatives aux événements filtrés.

Pour plus d'informations sur les autorisations, voir «Droits d'accès», à la page 295.

Création d'un filtre d'accès sur une vue

Pour créer un filtre d'accès, vous devez disposer d'une autorisation en lecture-écriture sur la vue.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail** et sélectionnez la vue sur laquelle portera le filtre.
2. Sélectionnez l'onglet **Filtres d'accès** pour afficher la liste des filtres déjà associés à la vue.
3. Cliquez sur **Créer un filtre d'accès** pour créer le nouveau filtre.
4. Ajoutez une table de correspondance à l'ensemble de tâches en cliquant sur **Ajouter des tables de correspondance**.
5. Dans la boîte de dialogue **Créer un filtre d'accès**, indiquez le nom du filtre, éventuellement accompagné d'une description.
6. Définissez la condition de filtre en suivant les instructions de la section «Conditions de filtre d'accès», à la page 3. Vous pouvez alors affecter ce filtre à des utilisateurs disposant d'un accès en lecture filtrée sur la vue.

Création de filtres d'accès basés sur des tables de correspondance

Il vous est possible de créer des filtres d'accès basés sur des tables de correspondance. Lorsque vous l'utilisez dans un filtre d'accès, une table de correspondance contient des informations compatibles avec le filtre, c'est-à-dire, des informations sur l'utilisateur en cours.

Avant de commencer

Vous devez disposer d'une autorisation de lecture et écriture sur le cube.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez le cube pour lequel vous voulez créer un filtre d'accès.
3. Cliquez sur l'onglet **Filtres d'accès** pour afficher la liste des filtres déjà associés au cube.
4. Cliquez sur **Créer un filtre d'accès**.
5. Dans la fenêtre Configurer le filtre du cube, indiquez le nom du filtre, éventuellement accompagné d'une description.
6. Cliquez sur **Ajouter un filtre**. Lorsque vous créez plusieurs filtres, une opération AND est exécutée sur les filtres afin de générer le résultat du filtre d'accès.
7. Choisissez une fonction.
8. Cliquez sur **Parcourir** pour sélectionner une table de correspondance et des colonnes.
9. Cliquez sur **Parcourir** pour sélectionner une dimension et un membre.
10. Affectez le filtre aux utilisateurs qui disposent d'un accès **Filtré / Lecture seule** sur le cube. Pour plus d'informations sur les effets de votre définition sur les conditions de filtre, voir «Conditions de filtre d'accès», à la page 3.

Création d'un filtre d'accès simple

Les filtres simples permettent de limiter les résultats de la vue pour n'accepter que les lignes qui répondent à la condition indiquée.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez le cube pour lequel vous voulez créer un filtre d'accès.
3. Cliquez sur l'onglet **Filtres d'accès** pour afficher la liste des filtres déjà associés au cube.
4. Cliquez sur **Créer un filtre d'accès**.
5. Sélectionnez **Utiliser des filtres simples**.
6. Cliquez sur **Parcourir** et sélectionnez la dimension à utiliser pour le filtre simple.
7. Sélectionnez un membre. Par exemple, si vous possédez une dimension d'emplacement, il se peut que l'un des membres soit une région ou un état.
8. Sélectionnez une valeur pour le filtre. Par exemple, si vous avez sélectionné Etat comme membre de la dimension d'emplacement, la valeur peut être Californie.
9. Cliquez sur l'option **Ajouter**.
10. Cliquez sur **Enregistrer**.
11. Affectez le filtre aux utilisateurs qui disposent d'un accès **Filtré / Lecture seule** sur le cube.

Affectation d'un filtre d'accès aux utilisateurs et aux rôles

Vous pouvez affecter un filtre d'accès à un utilisateur ou à un rôle.

Lorsque vous affectez plusieurs filtres, l'utilisateur voit les lignes remplissant l'une de ces conditions. Par exemple, lorsqu'un filtre n'affiche que les lignes associées au service d'un utilisateur, et qu'un autre filtre présente les lignes applicables à la région métier de cet utilisateur, ce dernier voit la ligne pour laquelle l'une ou l'autre condition est vérifiée.

Avant de commencer

Pour effectuer cette opération, vous devez disposer d'une autorisation en lecture et en écriture sur la vue ou sur le cube et les filtres doivent déjà avoir été définis.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez la vue pour laquelle vous souhaitez affecter l'accès en lecture seule filtré.
3. Cliquez sur **Activités** et sélectionnez **Droits**.
4. Sélectionnez un ou plusieurs utilisateurs ou rôles auxquels affecter l'accès filtré.
5. Cliquez sur **Modifier les droits**.
6. Choisissez l'accès **Filtré / Lecture seule**, puis un ou plusieurs filtres d'accès à utiliser.
7. Cliquez sur **OK**.

8. Cliquez sur **Terminé**.

Chapitre 3. Structure de l'adaptateur

La structure de l'adaptateur est un ensemble de composants que vous pouvez utiliser pour ajouter des agents externes à IBM Cognos Real-time Monitoring.

Certains agents sont étroitement liés à la source de données qu'ils utilisent. Par conséquent, il n'est pas possible pour Real-time Monitoring de communiquer avec la source en vue d'échanger des données.

La structure de l'adaptateur permet de connecter IBM Cognos Real-time Monitoring à des sources de données et à des applications externes via votre propre agent externe plutôt que l'un de ceux disponibles dans IBM Cognos Real-time Monitoring. Vous pouvez créer des adaptateurs externes configurables qui prennent en charge des opérations générales telles que le démarrage, l'arrêt et l'interrogation.

Lorsque vous implémentez un agent d'adaptateur externe, vous pouvez également créer un flux de données et des objets de table de correspondance qui utilisent votre agent.

La structure de l'adaptateur est constituée de plusieurs composants que vous implémentez pour connecter votre application externe à IBM Cognos Real-time Monitoring. Ces composants sont les suivants :

- Fichier de configuration de l'agent externe

  Le fichier de configuration de l'agent externe est un fichier XML qui définit la configuration de l'agent externe. Il spécifie le type de données que l'agent reçoit, ce qui détermine aussi les types de flux de données et les tables de correspondance mis à votre disposition une fois que vous avez configuré un agent externe. Pour plus d'informations, voir «Configuration du fichier de configuration», à la page 14.
- Agent externe

  Vous définissez l'agent externe dans IBM Cognos Real-time Monitoring de manière à vous connecter à votre application externe. Pour plus d'informations, voir «Création d'agents», à la page 28.
- Client externe

  Le client externe représente l'application externe qui adapte votre source de données aux exigences de Cognos Real-time Monitoring. Il est constitué de deux composants : la structure du client externe et le plug-in du client externe.
  - Structure du client externe

    Vous implémentez la structure du client externe de manière à gérer la réception des commandes et l'envoi des données à Cognos Real-time Monitoring. Celle-ci est fournie dans le fichier d'archive Java (JAR) qui contient les interfaces API destinées au plug-in du client externe.
  - Plug-in du client externe

    Vous créez le plug-in du client externe à l'aide des interfaces API de la structure du client externe. Ce plug-in extrait les données de votre source de données.

© Copyright IBM Corp. 2007, 2014

Création de la structure de l'adaptateur externe

Vous pouvez créer une structure permettant d'ajouter des agents externes à IBM Cognos Real-time Monitoring.

Procédure

1. Implémentez un plug-in client externe.

   Pour en savoir davantage, voir «Plug-in de l'adaptateur externe».
2. Créez un fichier XML AgentConfig basé sur votre plug-in.

   Pour en savoir davantage, voir «Configuration du fichier de configuration», à la page 14.
3. Intégrez le fichier XML ExternalAgentConfig à un fichier JAR ExternalAgentConfig.

   Pour en savoir davantage, voir «Fichier ConfigXML», à la page 15.
4. Téléchargez le fichier vers Cognos Real-time Monitoring Workbench en tant que fichier JAR.

   Pour en savoir davantage, voir «Dossier des fichiers JAR», à la page 16.
5. Enregistrez votre plug-in compilé avec le fichier JAR de l'adaptateur externe dans la machine à partir de laquelle le plug-in est en mesure d'accéder aux données appropriées. Incluez les classes de votre plug-in dans le chemin d'accès aux classes lors du démarrage de l'adaptateur externe.

   Pour en savoir davantage, voir «Compilation et démarrage d'une application externe», à la page 16.
6. Dans Cognos Real-time Monitoring Workbench, créez un agent avec le type que vous avez défini dans votre fichier XML ExternalAgentConfig.
7. Créez une table de correspondance ou de flux de données dans le plan de travail à l'aide de l'agent que vous avez créé.
8. Servez-vous des données stockées dans Cognos Real-time Monitoring pour générer des vues et des cubes.

Plug-in de l'adaptateur externe

Vous pouvez utiliser un plug-in existant, par exemple celui fourni avec IBM Cognos Real-time Monitoring ou en implémenter un nouveau.

Le plug-in implémente une classe qui prolonge la classe com.ibm.cognos.externaladapter.ExternalAdapterPlugin.

Cette classe extrait les données à partir du système source et les convertit en un format pris en charge par la structure de l'adaptateur externe.

Les méthodes ci-dessous doivent être implémentées dans la classe de plug-in. Un exemple est disponible dans : *emplacement_installation*/realtime/webcontent/samples/sdk/adapters/jdbc_sample/src/com/externalApplication/jdbc, où *emplacement_installation* désigne le répertoire d'installation d'IBM Cognos Business Intelligence.

Init()

Cette méthode est appelée lors de la création du flux de données de l'adaptateur externe. Elle comporte deux paramètres : un pour les détails sur l'agent et l'autre pour les détails sur le flux de données. Si une procédure doit être exécutée lors de l'initialisation du flux de données, cette méthode peut l'ajouter.

Avertissement : Aucune méthode n'est appelée lorsque l'agent est créé dans le plan de travail. La validation des données d'entrée de l'agent peut être effectuée à l'aide de la méthode testAgentCredentials() décrite ci-après.

Destroy()

Cette méthode est appelée lors de la suppression du flux de données. Elle permet de supprimer l'ensemble des données qui ont été créées lors de la méthode init().

testAgentCredentials()

Cette méthode est utilisée pour la fonctionnalité **Tester la connexion**. Elle permet de valider les données fournies par l'agent. Si l'agent ne requiert pas de validation, cette méthode doit être implémentée vide, ce qui renvoie une valeur par défaut True.

validateDataStream()

Cette méthode valide les données de flux de données que le plug-in reçoit. Le plug-in fournit une implémentation par défaut pour cette méthode, qui renvoie la valeur True. Par exemple, vous utilisez une application JDBC et vous indiquez un nom de table comme entrée. La validation de ce flux de données vérifie que le nom de la table est correct ou qu'il existe dans la base de données. Cette méthode est facultative.

getTabularData()

Cette méthode extrait des données à partir du flux de données tabulaires et les insère dans l'objet réponse dans un format reconnu par Cognos Real-time Monitoring. Cette méthode comporte deux paramètres : un pour les détails sur le flux de données et l'autre pour la réponse, qui est renvoyée vers l'application Cognos Real-time Monitoring.

L'appel IExternalAdapterTabularDataStreams getInputData() renvoie l'entrée spécifiée dans la zone **Entrée** de la source de données dans Cognos Real-time Monitoring Workbench. La méthode isIncrementalField() associée aux objets IExternalAdapterTabularDataStreamColumn indique s'il y a lieu de traiter la colonne en tant que zone d'incrémentation. Le plug-in conserve la valeur la plus grande produite par cette colonne. Lorsque le plug-in renvoie des données, il tient compte de cette valeur et de la valeur renvoyée par getInitialValue().

Chaque objet de cellule représente les colonnes séparées spécifiées lors de la création du flux de données. L'objet de classe est ExternalAdapterTabularDataStreamCellData. Il contient deux objets distincts : le type de données de la colonne et les données réelles.

getFlatFileData() et getXMLData()

Ces méthodes sont utilisées pour les flux de données de fichier à plat et XML.

Ces fichiers acceptent deux paramètres au même titre que la méthode getTabularData(). Toutefois, le format de l'objet réponse est différent. Au lieu d'envoyer une liste de lignes, ces deux fonctions renseignent une zone FileInputStream.

Exception.properties

Le fichier exception.properties affiche un message localisé détaillé de l'exception dans l'interface utilisateur.

Les classes d'exception, qui sont créées dans l'API, utilisent un paramètre de clé de message. Lorsqu'une clé est transmise à la classe d'exception, elle extrait d'abord le message du fichier exception.properties relatifs aux paramètres régionaux appropriés, puis l'affiche dans l'interface utilisateur.

Vous pouvez consulter un fichier de propriétés d'exception existant `exception_en.properties`. Par défaut, il se trouve dans le répertoire *`emplacement_installation`*`/realtime/webcontent/samples/sdk/adapters/template`, où *`emplacement_installation`* désigne le répertoire d'installation d'IBM Cognos BI.

Configuration du fichier de configuration

Le fichier de configuration détermine les types de transfert ainsi que les options disponibles dans la boîte de dialogue **Créer un agent**.

Vous pouvez avoir recours à un exemple de fichier de configuration existant, tel que `sampleAgentConfig1.xml`. Par défaut, il se trouve dans le répertoire *`emplacement_installation`*`/realtime/webcontent/samples/sdk/adapters/jdbc_sample/config`, où *`emplacement_installation`* désigne le répertoire d'installation d'IBM Cognos BI.

Le fichier de configuration est un fichier XML qui contient les éléments suivants :

- name : identificateur de l'agent.
- nameKey : nom du type de l'agent. Il s'agit d'une clé localisable pour la chaîne qui s'affiche dans la boîte de dialogue **Créer un agent**.
- supportsIncrementingField : indicateur booléen qui spécifie si l'adaptateur prend en charge une zone d'incrémentation.
- transportType : liste séparée par des virgules qui indique le format des données. Ceci détermine les éléments fournis par l'interface utilisateur.
  - TABULAR
  - DELIMITED
  - FIXEDWIDTH
  - XMLFILE
- ExternalAgentVersionSupported : version du plug-in de l'adaptateur externe.

Fichier ConfigXML

Le fichier ConfigXML est compressé dans un fichier d'archive Java (JAR).

Le fichier JAR utilise une structure de dossier similaire à celle ci-dessous :

Création d'un fichier JAR

Tableau 1. Composants du fichier JAR

Composant de fichier JAR	Description
Fichiers de configuration	Lorsque vous téléchargez le fichier JAR, IBM Cognos Real-time Monitoring convertit les fichiers XML en objets ExternalAgentConfig. Vous pouvez voir ces objets en tant que dépendances du fichier JAR téléchargé.
Fichier manifeste	Ce fichier décrit le contenu du fichier JAR dans Cognos Real-time Monitoring. Il contient les informations nécessaires aux fichiers de configuration et au regroupement de ressources.
Fichiers de ressources	Ces fichiers fournissent les chaînes localisées. Cette chaîne correspond au nom d'affichage de l'objet ExternalAgentConfig. La chaîne localisée est fournie à Cognos Real-time Monitoring en fonction des paramètres régionaux associés à l'application. Les clés du fichier de ressource sont fournies dans le fichier de configuration comme nom de clé. Lorsque l'objet de configuration de l'agent est créé, à l'aide des paramètres régionaux et de la clé de nom, la chaîne réelle du regroupement de ressources est extraite et affichée à l'écran.

Vous pouvez utiliser un outil de conception Java, notamment Ant, pour créer le fichier JAR. Ant vous permet de compiler et de générer l'adaptateur, de compresser les fichiers et de créer un dossier JAR.

Si vous souhaitez automatiser ce processus à l'aide de l'outil Ant, vous pouvez vous servir du fichier exemple `build.xml` situé dans : *`emplacement_installation`*`\realtime\webcontent\samples\sdk\fileagent\com\ibm\cognos\externaladapter`, où *`emplacement_installation`* désigne le répertoire d'installation d'IBM Cognos BI.

Vous exécutez la cible *build-config-jar* pour générer le fichier JAR ConfigXML.

Si vous préférez effectuer cette opération manuellement, vous devez au préalable avoir créé un dossier et avoir reproduit la même structure de répertoires que celle

illustrée dans le diagramme ci-dessus. Vous vous placez ensuite dans le dossier que vous avez créé et vous exécutez la commande permettant de créer le fichier JAR.

Lorsque vous créez le fichier JAR, n'ajoutez pas le dossier racine au fichier JAR.

- Utilisez la commande DOS CD pour accéder au dossier racine.

  Par exemple, si le nom du dossier racine est `AgentConfig` :

  ```
  CD <path of the root directory> AgentConfig
  ```

- Exécutez ensuite la commande :

  ```
  jar  cvf * <jar file name>
  ```

  'v' est facultatif.

Dossier des fichiers JAR

Les fichiers JAR contiennent un ou plusieurs programmes Java.

Le dossier Fichiers JAR permet d'accéder aux fichiers JAR disponibles pour IBM Cognos Real-time Monitoring. Sélectionnez le dossier pour afficher la liste des fichiers JAR déjà installés et en charger de nouveaux. Dans le dossier, sélectionnez un fichier JAR pour visualiser ses fonctions et les objets qui en dépendent.

Procédure

1. Dans le plan de travail, cliquez sur le dossier **Fichiers JAR**.
2. Cliquez sur **Nouveau fichier JAR**.
3. Saisissez un nom pour le fichier JAR dans la zone **Nom** et, éventuellement, une description.
4. Cliquez sur **Parcourir** pour rechercher le fichier dans la zone **Chemin d'accès**.
5. Cliquez sur **Enregistrer**.

   Le fichier JAR est désormais disponible sur le système.

Compilation et démarrage d'une application externe

Le fichier `build.xml` permet de compiler le plug-in de l'adaptateur externe.

La valeur de classe principale du fichier `build.xml` est toujours com.ibm.cognos.externaladapter.ExternalAdapterServer, car elle démarre le serveur externe. Le fichier ExternalAdapter.jar contient les classes d'adaptateur externe nécessaires à la compilation de la classe de plug-in.

Démarrage du processus d'adaptateur externe

Pour exécuter l'agent externe, vous devez appeler le processus Java avec la classe com.ibm.cognos.externaladapter.ExternalAdapterServer et un chemin d'accès aux classes correctement configuré. Ce chemin doit comporter le fichier `ExternalAdapter.jar`, qui contient les classes requises pour l'implémentation de votre plug-in, tous les fichiers JAR dont dépend votre implémentation, le fichier `log4j-1.2.7.jar`, ainsi que le répertoire dans lequel résident les fichiers `config.properties` et `exception*.properties`. Par exemple : `java -classpath lib\ExternalAdapter.jar;lib\`*`your plug-in JAR`*`;lib\log4j-1.2.7.jar;lib\SampleApplication.jar;. com.ibm.cognos.externaladapter.ExternalAdapterServer`

Avertissement : Si vous utilisez le fichier modèle `build.xml`, la cible d'exécution configure le chemin d'accès aux classes et démarre l'adaptateur externe.

Valeurs des paramètres du fichier build.xml

- Propriété relative au répertoire qui contient l'ensemble des fichiers de génération :

  ```
  <property name="build.dir" value="ExternalAdaperSampleApplication"/>
  ```

- Propriété relative au nom et à la destination du modèle de fichier d'application exécutable :

  ```
  <property name="externalapplicationjar" value="SampleApplication.jar"/>
  ```

- Propriété relative au répertoire source qui contient l'ensemble des fichiers Java créés pour l'implémentation du plug-in, par exemple, la classe de plug-in :

  ```
  <property name="src" value="."/>
  ```

- Propriété relative au répertoire qui contient l'ensemble des classes compilées :

  ```
  <property name="classes" value="${build.dir}"/>
  ```

config.properties

Un fichier de configuration est requis pour exécuter les applications externes. La classe du serveur s'appuie sur ce fichier pour charger les propriétés de configuration spécifiques, telles que la propriété PORT.

La propriété PORT indique le numéro de port sur lequel l'application externe s'exécute et communique avec IBM Cognos Real-time Monitoring.

Pour obtenir un exemple de fichier `config.properties`, voir : *emplacement_installation*`\realtime\webcontent\samples\sdk\fileagent\com\ibm\cognos\externaladapter`, où *emplacement_installation* indique l'emplacement d'installation d'IBM Cognos BI.

Fichiers JAR supplémentaires

L'application externe requiert des fichiers JAR supplémentaires lors de l'amorçage.

Elle requiert les fichiers `log4j-1.2.7.jar`, `ExternalAdapter.jar` et `SampleApplication.jar`. Si le démarrage de l'application nécessite des fichiers JAR supplémentaires, notamment les fichiers JAR du pilote JDBC, vous devez les ajouter au chemin d'accès aux classes lorsque vous exécutez l'adaptateur externe.

Agents externes

Il vous est possible de créer un agent externe en fonction de l'objet ExternalAgentConfig du fichier JAR téléchargé.

Créez un agent. Pour plus d'informations, voir «Création d'agents», à la page 28.

Dans la liste déroulante **Type d'agent**, recherchez la chaîne localisée de l'objet ExternalAgentTypeNameKey, comme indiqué dans l'objet ExternalAgentConfig. Pour plus d'informations, voir «Configuration du fichier de configuration», à la page 14.

Une fois que vous avez sélectionné l'agent, sous **Détails de l'agent**, indiquez les attributs suivants :

Tableau 2. Détails de l'agent externe

Détails de l'agent	Description
Nom d'hôte externe	Adresse IP de la machine sur laquelle l'application externe s'exécute.
Port externe	Numéro de port sur lequel l'application externe est en cours d'exécution, tel qu'indiqué dans le fichier `config.properties`. Pour plus d'informations, voir «Compilation et démarrage d'une application externe», à la page 16
Entrée	Propriétés de connexion requises par l'application externe. Cette même valeur est attribuée aux fonctions testAgentCredentials() et init() dans la classe de plug-in. Pour plus d'informations, voir «Plug-in de l'adaptateur externe», à la page 12.

Flux de données externes

Une fois que vous avez implémenté votre agent externe dans IBM Cognos Real-time Monitoring, vous pouvez configurer les flux de données mis à la disposition de votre agent externe.

Les types de flux de données que vous créez dépendent des éléments transportType que vous spécifiez dans le fichier de configuration de votre agent externe. Pour plus d'informations, voir «Configuration du fichier de configuration», à la page 14.

Pour en savoir plus sur les flux de données, voir Chapitre 10, «Flux de données», à la page 85.

Vous avez la possibilité de créer quatre types de flux de données, en fonction du type de transfert pris en charge par votre agent externe. Les types de flux de données sont les suivants :

- Tabulaire
- Fichier à plat - Délimité
- Fichier à plat - Largeur fixe
- Fichier XML

Vous sélectionnez le type dans la fenêtre qui s'affiche une fois que vous avez sélectionné l'agent externe.

Flux de données tabulaires

Un flux de données tabulaires est similaire à un flux de données HTTP Post. Un flux de données tabulaires possède les attributs décrits dans le tableau ci-après. Vous pouvez configurer un flux de données tabulaires dans Cognos Real-time Monitoring Workbench si le type de transfert Tabulaire est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de flux de données. Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-après.

Chaque zone du message peut correspondre à une zone simple mappée directement à une colonne de la table de flux de données, ou à une zone plus complexe (zone de fichier à plat) qui comporte plusieurs zones mappées individuellement à des colonnes de la table. Les zones complexes sont traitées en tant que fichiers à plat au format CSV ou à largeur fixe.

Flux de données pour les fichiers à plat délimités

Le flux de données des fichiers à plat délimités reçoit des données dans lesquelles chaque zone (colonne) est séparée par un caractère, une virgule le plus souvent. Vous pouvez configurer un flux de données pour les fichiers à plat délimités dans Cognos Real-time Monitoring Workbench si le type de transfert Délimité est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de flux de données. Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-après.

Flux de données pour les fichiers à plat à largeur fixe

Le flux de données des fichiers à plat à largeur fixe reçoit des données dans lesquelles chaque zone (colonne) présente la même largeur prédéfinie sur chaque ligne du fichier, comme dans une table de feuille de calcul. Vous pouvez configurer un flux de données pour les fichiers à plat à largeur fixe dans Cognos Real-time Monitoring Workbench si le type de transfert Largeur fixe est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de flux de données. Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-après.

Flux de données XML

Le flux de données XML reçoit des données au format XML. Vous pouvez configurer un flux de données XML externe dans IBM Cognos Real-time Monitoring Workbench si le type de transfert Fichier XML est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de flux de données. Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-après.

Intervalle de suppression de l'état et interrogation

Tous les flux de données externes possèdent des onglets **Intervalle de suppression de l'état** et **Interrogation**.

Pour en savoir plus sur l'onglet **Intervalle de suppression de l'état**, voir «Intervalle de suppression de l'état», à la page 61.

L'onglet **Interrogation** permet d'indiquer la fréquence d'interrogation de la source externe à la recherche de nouvelles données.

Tableau 3. Attributs des flux de données tabulaires

Paramètre	Description
Interroger selon un intervalle	Indique la fréquence d'exécution de la requête.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive le flux de données (interrompt l'interrogation) lorsque le nombre d'erreurs consécutives atteint le seuil spécifié. Par défaut, l'interrogation s'arrête après cinq erreurs consécutives. Pour activer de nouveau le flux de données, modifiez son paramètre. Spécifiez la valeur zéro (0) pour que le flux de données ne soit jamais désactivé.
Zone d'incrémentation	Identifie la colonne de la table source, qui contient une valeur incrémentée automatiquement à chaque nouvel événement. Par exemple, si la table interrogée contient des ID uniques croissants, cette zone est celle qui est utilisée par la requête appliquant la logique où "l'ID est supérieur à l'ID maximum de la dernière requête". La zone d'incrémentation est disponible uniquement si l'option supportsIncrementingField est définie sur true dans le fichier de configuration de l'agent externe.
Valeur initiale	Valeurs de début attribuées à la zone d'incrémentation. Toutefois, cette valeur initiale n'est pas incluse dans le résultat. La première valeur incluse dans le résultat sera la valeur supérieure la plus proche à la valeur initiale. Par exemple, si vous indiquez la valeur initiale 500, la première valeur incluse sera 501. Pour les requêtes suivantes, cette valeur doit être supérieure à celle de la zone d'incrémentation renvoyée par la dernière requête.

Tables de correspondance externes

Une fois que vous avez implémenté votre agent externe dans IBM Cognos Real-time Monitoring, vous pouvez configurer les tables de correspondance mises à la disposition de votre agent externe.

Les types de table de correspondance que vous créez dépendent des éléments transportType que vous spécifiez dans le fichier de configuration de votre agent externe. Pour en savoir plus sur le fichier de configuration, voir «Configuration du fichier de configuration», à la page 14.

Pour en savoir plus sur les tables de correspondance, voir Chapitre 20, «Tables de correspondance», à la page 265.

Vous avez la possibilité de créer quatre types de tables de correspondance, en fonction du type de transfert pris en charge par votre agent externe. Les types de table de correspondance sont les suivants :

- Tabulaire
- Fichier à plat - Délimité
- Fichier à plat - Largeur fixe
- Fichier XML

Vous sélectionnez le type dans la fenêtre qui s'affiche une fois que vous avez sélectionné l'agent externe.

Tables de correspondance tabulaires

Une table de correspondance est similaire à la table de correspondance d'un système de fichiers. Une table de correspondance tabulaire possède les attributs décrits dans le tableau ci-après. Vous pouvez configurer une table de correspondance tabulaire dans IBM Cognos Real-time Monitoring Workbench si le type de transfert Tabulaire est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de correspondance. Chaque colonne de la table de correspondance possède les attributs décrits dans le tableau ci-après.

Chaque zone du message peut correspondre à une zone simple mappée directement à une colonne de la table de correspondance, ou à une zone plus complexe (zone de fichier à plat) qui comporte plusieurs zones mappées individuellement à des colonnes de la table. Les zones complexes sont traitées en tant que fichiers à plat au format CSV ou à largeur fixe.

Tables de correspondance pour les fichiers à plat délimités

La table de correspondance associée aux fichiers à plat délimités reçoit des données dans lesquelles chaque zone (colonne) est séparée par un caractère, une virgule le plus souvent. Vous pouvez configurer une table de correspondance pour les fichiers à plat délimités dans IBM Cognos Real-time Monitoring Workbench si le type de transfert Délimité est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de correspondance. Chaque colonne de la table de correspondance possède les attributs décrits dans le tableau ci-après.

Tables de correspondance pour les fichiers à plat à largeur fixe

La table de correspondance associée aux fichiers à plat à largeur fixe reçoit des données dans lesquelles chaque zone (colonne) présente la même largeur prédéfinie sur chaque ligne de fichier, comme dans une table de feuille de calcul. Vous pouvez configurer une table de correspondance pour les fichiers à plat à largeur fixe dans IBM Cognos Real-time Monitoring Workbench si le type de transfert Largeur fixe est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de correspondance. Chaque colonne de la table de correspondance possède les attributs décrits dans le tableau ci-après.

Tables de correspondance pour les données XML

La table de correspondance XML reçoit des données au format XML. Vous pouvez configurer une table de correspondance XML externe dans IBM Cognos Real-time Monitoring Workbench si le type de transfert Fichier XML est spécifié dans le fichier de configuration de votre agent externe.

Les zones de l'onglet **Informations sur la colonne** indiquent comment mapper les zones de la source de données en colonnes dans la table de correspondance. Chaque colonne de la table de correspondance possède les attributs décrits dans le tableau ci-après.

Mise en cache des données et calendrier d'invalidation

Chaque type de table de correspondance externe comporte un onglet **Mise en cache des données** et un onglet **Calendrier d'invalidation**.

Si vous envisagez d'utiliser une bibliothèque JDBC pour votre cache de données, vous aurez besoin d'un agent JDBC.

L'onglet **Mise en cache des données** propose les options de mise en cache suivantes :

- **Type de cache**

  Vous avez le choix entre la mise en cache en mémoire ou la bibliothèque JDBC.

  – **En mémoire**

    Lorsque vous activez la mise en cache en mémoire, la table de correspondance externe tout entière est mise en cache en mémoire à partir de la source de données externe et devient une solution de rechange pour la source de requête externe. Cela revient au même que si vous sélectionnez la mise en cache prefetch pour les sources de la table de correspondance à partir des agents JDBC et de service Web, comme décrit dans Mise en cache prefetch.

    Si vous sélectionnez cette option, l'option **Bibliothèque JDBC** n'est pas disponible et la zone **Agent JDBC** apparaît grisée.

  – **Bibliothèque JDBC**

    Lorsque vous sélectionnez **Bibliothèque JDBC**, les données de la table de correspondance sont extraites vers une base de données locale, comme spécifié par l'agent JDBC.

    La liste déroulante **Agent JDBC** présente les agents disponibles. Sélectionnez l'agent à utiliser pour la base de données locale.

  Extraire et stocker au redémarrage

  Cette option indique que lorsque vous redémarrez IBM Cognos Real-time Monitoring, le système récupère toujours les données de la source de requête externe et les stocke dans la base de données locale. Si une bibliothèque locale déjà alimentée existe, elle est utilisée jusqu'au prochain calendrier d'invalidation. Si aucune bibliothèque locale n'a déjà reçu de données, la source externe est interrogée dès le redémarrage.

- **Recourir à une source externe**

  Cette option indique que si le système ne parvient pas à accéder à la base de données locale, il utilise les données provenant de la base de données externe. Cette option n'est pas disponible pour les tables de correspondance basées sur un agent de système de fichiers.**Nombre de résultats à mettre en cache**

  Cette option détermine le nombre d'ensembles de résultats à mettre en cache. Chaque ensemble de résultats peut contenir une ou plusieurs lignes de données de table de correspondance liées au flux de données. Par exemple, si trois requêtes sont effectuées et que chaque ensemble de résultats contient cinq lignes, 15 lignes sont stockées en cache.

La calendrier d'invalidation identifie le moment opportun pour invalider le cache et supprimer toutes les informations contenues dans le cache. Pour créer un calendrier d'invalidation, cliquez sur le bouton **Ajouter un calendrier**. Les actions suivantes sont disponibles :

- Supprimer le cache à une heure précise ou au cours d'un intervalle de temps donné.
- Supprimer le cache quotidiennement, un jour spécifique de la semaine ou un ou plusieurs jours spécifiques du mois.
- Supprimer le cache au cours d'un ou de plusieurs mois spécifiques.

Par exemple, pour planifier la suppression du cache trimestriellement le 30 du mois à 22 heures, procédez comme suit :

- Sélectionnez **Ajouter une heure précise** et définissez **Programmer à** sur 22 heures.
- Sélectionnez **Chacun des jours du mois sélectionnés ci-dessous** et choisissez **30**.
- Sélectionnez les mois de mars, juin, septembre et décembre.

Chapitre 4. Agents

Les agents sont des processus qui récupèrent des informations à partir de sources externes lorsqu'une mise à jour se produit.

Lorsqu'un agent localise de nouvelles informations, il transmet ces données aux tables de flux de données et de correspondance pour permettre leur utilisation par les vues.

Figure 1. Diagramme de flux de travaux de l'activité d'un agent

Un même agent peut prendre en charge plusieurs tables de flux de données ou de correspondance.

Alors qu'un agent sait comment communiquer avec une source externe, les tables de flux de données et de correspondance savent déterminer les informations nécessaires. La plupart des flux de données et des tables de correspondance définissent les détails qui précisent à l'agent quelles informations rechercher.

Sources externes

Les agents permettent de traiter différentes sources d'informations externes.

Les informations sont insérées dans le système lors de la mise à jour des informations ou à la suite d'une requête provenant d'une base de données ou d'un fichier texte. Les informations destinées aux tables de correspondance sont toujours extraites à partir de la source.

Les sections suivantes répertorient les sources disponibles et identifient les agents source pris en charge. Chaque tableau indique si l'agent transmet ou extrait des informations de mise à jour pour un flux de données ou une table de correspondance.

Connectivité JDBC à une base de données

Lorsque JDBC (Java Database Connectivity) fait appel à une source externe, celle-ci provient généralement d'une base de données relationnelle (SGBDR).

© Copyright IBM Corp. 2007, 2014

Tableau 4. Envoi ou extraction JDBC

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
JDBC (voir Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227)	Non	Oui	Oui

Java Messaging Service à partir d'une application Java

Lorsque le service JMS est une source externe, il s'agit d'un service JMS à partir d'une application Java.

Tableau 5. Envoi ou extraction JMS

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
File d'attente JMS (voir Chapitre 17, «Java Messaging Service (JMS)», à la page 221)	Oui	Non	Non
Rubrique JMS (voir Chapitre 17, «Java Messaging Service (JMS)», à la page 221)	Oui	Non	Non

Fichiers texte

Les fichiers texte sont des fichiers à plat. Ils peuvent faire appel à un agent de fichier à plat ou à un agent de système de fichiers.

Tableau 6. Envoi ou extraction de fichiers texte

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Fichiers à plat (voir Chapitre 13, «Fichiers à plat», à la page 103)	Non	Oui	Non
Système de fichiers (voir Chapitre 12, «Système de fichiers Cognos Real-time Monitoring», à la page 97)	Non	Non	Oui

TIBCO Rendezvous (RV)

Lorsque l'application TIBCO RV est une source externe, cette source provient d'une application métier qui utilise les flux de messages TIBCO.

Tableau 7. Envoi ou extraction TIBCO Rendezvous

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
TIBCO Rendezvous (voir Chapitre 33, «TIBCO Rendezvous» , à la page 349)	Oui	Non	Non

Service Web

Lorsqu'un service Web est une source externe, cette source provient d'une application Web via une connexion HTTP.

Tableau 8. Envoi ou extraction de service Web

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Services Web (voir «Processus externes des services Web», à la page 386)	Oui	Non	Oui

Connectivité SAP

Lorsque la source externe est une base de données pour système SAP, l'agent ERP permet d'accéder au système SAP en tant que type de source par le biais d'un agent JDBC de métadonnées SAP.

Tableau 9. Envoi ou extraction SAP

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
ERP (voir Chapitre 30, «Connectivité SAP», à la page 333)	Non	Oui	Non

Système Salesforce

Lorsque la source externe est un système Salesforce, la source correspond aux tables Salesforce.

Tableau 10. Envoi ou extraction Salesforce

Agent	Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Salesforce (voir Chapitre 29, «Salesforce», à la page 327)	Non	Oui	Oui

Création d'agents

Vous créez des agents afin de récupérer des informations à partir de sources externes lorsqu'une mise à jour de ces informations se produit.

Conseil :

Utilisez les agents JDBC basés sur la source de données et non sur l'URL. Les agents basés sur la source (ou JNDI) autorisent le regroupement de connexions, mais pas les agents basés sur l'adresse URL.

Assurez-vous que le nombre de connexions du pool de connexions JNDI est suffisant. La valeur par défaut est généralement 5 ou 20, ce qui est trop faible lorsque toutes les tables de correspondance et les flux de données partagent le même agent. Augmentez le nombre de connexions (valeur entre 50 et 100).

Utilisez le paramètre **Nombre maximal de lignes par requête** pour qu'un agent JDBC propage la charge. Si ce paramètre n'est pas défini, une requête via un agent JDBC extrait tout l'ensemble de résultats en même temps. Si cet ensemble contient un grand nombre de données, le système ne traite aucune donnée tant que l'ensemble tout entier n'a pas été extrait. Si ce paramètre est défini, un nombre limité de lignes est extrait de la base de données et traité. Les lignes restantes sont extraites au cours du prochain cycle d'extraction. Les cycles d'extraction sont déterminés par interrogation. L'interrogation indique à l'objet la fréquence à laquelle interroger le SGBD à la recherche de nouveaux événements. Une technique efficace consiste à interroger le SGBD de manière fréquente, par exemple 5 000 lignes par interrogation. N'utilisez pas cette technique pour les agents de table de correspondance, car toutes les lignes ne sont pas chargées.

Utilisez des agents différents pour les tables de correspondance et les flux de données. Assurez-vous que la taille du pool de connexions de la table de correspondance est suffisante pour traiter chaque table de correspondance créée. Si la mise en cache prefetch n'est pas activée pour la table de correspondance, cette dernière doit disposer d'un agent dédié afin d'empêcher que d'autres tables de correspondance ou flux de données ne puissent accéder à la base de données.

Pour les implémentations JBoss, vous devez configurer une définition JNDI correspondante dans un fichier celequest_context-ds.xml. Pour plus d'informations, reportez-vous aux instructions de déploiement JBoss du guide d'installation.

Pour plus d'informations sur les types d'agents, voir «Sources externes», à la page 25.

Si vous faites appel à un agent externe, le nom de votre agent s'affiche dans la liste des types d'agents disponibles.

Pour plus d'informations sur la création d'un agent ERP, voir «Création d'un agent SAP», à la page 333. Pour plus d'informations sur la création d'un agent JDBC, voir Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227. Pour plus d'informations sur la création d'un agent Salesforce, voir Chapitre 29, «Salesforce», à la page 327.

Pour plus d'informations sur les types de sources, voir les rubriques suivantes :

- «Agents de fichier à plat», à la page 30
- «Création d'un agent de système de fichiers», à la page 34

- «Agents JDBC (Java Database Connectivity)», à la page 238
- «Agents de file d'attente JMS (Java Messaging Service)», à la page 35
- «Agents de rubrique JMS», à la page 38
- «Agents TIBCO Rendezvous», à la page 40
- «Agents de services Web», à la page 41

Avant de commencer

Pour pouvoir créer un agent, vous devez disposer d'une autorisation de création sur les agents ainsi que des spécifications de connexion pour ce type d'agent. Pour plus d'informations sur les autorisations, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Choisissez un type d'agent.
5. Renseignez les détails relatifs au type de source spécifique.
6. Enregistrez l'objet avec l'état activé.

Modification des agents

Il se peut que vous ayez besoin de modifier un agent existant.

Avant de commencer

Avant de modifier un agent, vous devez disposer des droits de lecture et d'écriture pour cet agent. Pour plus d'informations sur les autorisations, voir «Droits d'accès» , à la page 295.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez l'agent de l'une des manières suivantes :
   - Dans la sous-fenêtre Tables et vues, recherchez l'agent à modifier et cliquez deux fois sur son nom.
   - Sélectionnez l'agent dans la sous-fenêtre droite, cliquez sur **Activités**, puis sur **Modifier**.
3.
4. Si vous avez cliqué deux fois sur le nom d'un agent pour le sélectionner, cliquez sur **Modifier** en haut de l'écran **Dépendances**. Sinon, passez à l'étape 5.
5. Renseignez les détails relatifs au type d'agent spécifique.
6. Enregistrez l'objet avec l'état activé.

Agents de fichier à plat

Un agent de fichier à plat extrait des données à partir d'un fichier texte.

Tableau 11. Envoi ou extraction de fichier à plat

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Non	Oui	Non

Traitement des fichiers

L'agent de fichier à plat recherche des fichiers à un emplacement spécifié selon une fréquence définie. Vous pouvez utiliser un astérisque (*) ou un point d'interrogation (?) comme caractère générique dans le nom des fichiers à rechercher. Lorsque l'agent localise un fichier, il extrait les données de la table de correspondance ou de l'événement, puis supprime, déplace ou renomme le fichier source. S'il détecte plusieurs fichiers à l'emplacement défini, il les traite dans l'ordre alphabétique.

Prérequis

Pour créer un agent de fichier à plat, vous devez disposer des éléments suivants :

- Autorisation de création sur les agents. Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.
- Programme d'agent de fichier en cours d'exécution. Pour en savoir davantage, voir «Programme de l'agent de fichier», à la page 31.

Chaque agent de fichier possède les attributs suivants :

Tableau 12. Attributs d'agent de fichier

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents. Pour les données issues de fichiers à plat, il s'agit du nom défini par l'élément agentName dans le fichier de configuration «TestAgent.xml», à la page 31. Pour en savoir davantage, voir «Programme de l'agent de fichier», à la page 31.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.

Création d'un agent de fichier à plat

Utilisez la procédure suivante pour créer un agent de fichier à plat.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**, puis cliquez sur le bouton **Activités**.
2. Sélectionnez **Créer**, puis **Agent** dans la liste déroulante.
3. Choisissez le type d'agent **Fichier à plat**.
4. Renseignez les zones qui définissent les attributs de l'agent.
5. Enregistrez l'agent avec l'état activé pour qu'il commence immédiatement à rechercher des événements.

Programme de l'agent de fichier

Le programme de l'agent de fichier est un programme Java autonome s'exécutant sur un hôte qui collecte des données à partir d'un fichier texte. Cet hôte peut différer de l'hôte du serveur IBM Cognos Real-time Monitoring. Lorsque le programme de l'agent de fichier détecte des données, il les transmet aux serveurs Cognos Real-time Monitoring pour traitement.

Le programme de l'agent de fichier ne s'applique pas aux données de la table de correspondance à partir d'un fichier texte.

L'agent comporte deux fichiers de configuration XML :

- `TestAgent.xml` (basé sur `VCAgent.xsd`) : Définit les informations de connexion, telles que la procédure pour localiser les serveurs Cognos Real-time Monitoring et la façon dont ces serveurs peuvent identifier l'agent de fichier.
- `FileAgent.xml` (basé sur `FileAgent.xsd`) : Identifie le fichier texte ainsi que la procédure à suivre avec le fichier à l'issue du téléchargement de ses données.

Pour lancer l'agent, exécutez le fichier `cqagent.jar` dans Java, puis transmettez le fichier de configuration `TestAgent.xml` sous la forme d'un argument de la façon suivante :

```
java -jar ...\cqagent.jar TestAgent.xml
```

Vous pouvez également identifier le répertoire du fichier de journalisation et le niveau de journalisation en définissant les propriétés correspondantes. L'exemple ci-après définit le niveau de journalisation de tous les messages :

```
java "-Dcom.cognos.obi.property.Logging Directory=C:\logs\agents"
     "-Dcom.cognos.obi.property.Detailed Log File Level=All"
      -jar ...\cqagent.jar TestAgent.xml
```

TestAgent.xml

Cette section décrit les attributs et les éléments de configuration du fichier `TestAgent.xml`.

Tableau 13. Attributs TestAgent.xml

Attribut	Description
serverPort (attribut)	(Facultatif : la valeur par défaut est 80) Port HTTP du serveur d'applications exécutant les serveurs Cognos Real-time Monitoring et utilisé par l'agent pour communiquer avec le serveur. Ce port est identique à celui permettant aux utilisateurs de se connecter à Cognos Real-time Monitoring Workbench.

Tableau 13. Attributs TestAgent.xml (suite)

Attribut	Description
pingInterval (attribut)	(Facultatif : la valeur par défaut est 20 secondes) Fréquence à laquelle l'agent vérifie si le serveur Cognos Real-time Monitoring est en cours d'exécution. Lorsque le serveur n'est pas actif, l'agent ne regroupe pas les événements.
agentName	(Obligatoire) Identifie l'agent et indique le nom à utiliser lors de la création de l'agent dans la **console d'administration**. Ce nom doit être unique parmi les agents. Pour plus de détails, voir «Espace-noms d'objets», à la page 281.
serverHost	(Obligatoire) Nom de la machine hôte exécutant les serveurs Cognos Real-time Monitoring. Si ces serveurs s'exécutent sur la même machine que l'agent de fichier, indiquez le nom d'hôte localhost.
agentImplClass	(Obligatoire) Classe d'implémentation de l'agent. Cette valeur ne doit pas être modifiée. La classe com.cognos.obi.agent.FileAgent est requise pour l'instant.
agentImplConfigFile	(Obligatoire) Identifie le fichier de configuration nécessaire à l'implémentation (composant du fichier texte), généralement `FileAgent.xml`. Pour plus d'informations, voir «FileAgent.xml», à la page 33.
agentPort	(Obligatoire) Port utilisé pour communiquer avec l'agent sur l'hôte. Ce port permet au serveur de communiquer à l'agent les changements d'état de désactivation et d'activation. Utilisez n'importe quel numéro de port valide, par exemple 5050.
pollingInterval	(Obligatoire) Fréquence (en secondes) de recherche de nouveaux événements.
loggingDirectory	(Facultatif : la valeur par défaut correspond au répertoire des fichiers de configuration). Répertoire dans lequel stocker les informations de fichier. Le nom de fichier journal est `agentName.log`.
serverPath	Définissez cet attribut de manière qu'il pointe vers l'adresse URL du servlet ffs. Il peut ainsi pointer vers le serveur d'applications ou le serveur Web. Si cqagent est en mesure d'accéder au serveur Web d'application, remplacez l'attribut par /realtime/ffs. Si vous faites appel à l'interface du serveur Web, remplacez l'attribut par /cognos/realtime/ffs.

Exemple

L'exemple ci-après attribue à l'agent d'événements le nom "orderStatusEvent", choisit FileAgent.xml en tant que fichier de configuration de l'implémentation et configure le port du serveur sur 8080 :

```
<?xml version="1.0" encoding="UTF-8" ?>
<VCAgent xmlns="http://cognos.obi.com/5"
   xmlns:xsi="http://www.w3.org/2001/xmlSchema-instance"
   xsi:schemaLocation="http://cognos.obi.com/5 VCAgent.xsd"
serverPort="80"
>
<agentName>flatFileEvent</agentName>
<serverHost>localhost</serverHost>
   <agentImplClass>com.cognos.obi.agent.FileAgent</agentImplClass>
```

```
<agentImplConfigFile>FileAgent.xml</agentImplConfigFile>
<agentPort>5050</agentPort>
<pollingInterval>20</pollingInterval>
</VCAgent>
```

FileAgent.xml

Ce fichier configure le composant de fichier texte (l'implémentation) de l'agent de fichier.

Le nom et l'emplacement de ce fichier sont identifiés dans le fichier `TestAgent.xml` (voir «TestAgent.xml», à la page 31) qui se trouve généralement dans le même répertoire. Ce fichier de configuration comporte quatre éléments de configuration, bien que la plupart des configurations n'utilisent que les éléments <filename> et <fileDisposal>.

Eléments

Le fichier comporte les éléments de configuration suivants.

Tableau 14. Eléments du fichier FileAgent.xml

Elément	Description
filename	Nom et emplacement du fichier texte source contenant les événements. Ce fichier est supposé se trouver dans le même répertoire que le fichier de configuration, sauf si vous identifiez un autre emplacement dans le nom de fichier. Vous pouvez spécifier un chemin d'accès relatif ou complet. Le nom du fichier peut inclure les caractères génériques * et ? . Sur les systèmes UNIX, utilisez une barre oblique pour séparer les noms des répertoires ; par exemple data_streams/file*.txt. Sur les systèmes Microsoft Windows, séparez les noms des répertoires à l'aide de deux barres obliques inversées ; par exemple data_streams\\file*.txt.
type	Identifie la source en tant que STREAM (Flux) ou FILE (Fichier). Utilisez FILE si la totalité du fichier texte doit être téléchargée de façon atomique (tout ou rien), par exemple dans le cas d'un fichier XML. Dans le cas contraire, utilisez STREAM pour télécharger des lignes sous forme de lots définis par l'élément buffersize.
buffersize	(Facultatif : valeur par défaut = 4,000+EOL) Nombre de caractères à stocker en mémoire tampon ou à envoyer par lots au serveur. Utilisez cette valeur si vous souhaitez éviter de télécharger une quantité de texte volumineuse en une seule fois.
fileDisposal	(Facultatif) Définit la procédure à suivre avec le fichier source après le téléchargement de ses données. Les options possibles sont les suivantes : **Delete** (option par défaut), qui supprime le fichier à l'issue du téléchargement ; **Move**, qui déplace le fichier vers un répertoire spécifié par l'attribut target ; et **Rename**, qui renomme le fichier en ajoutant l'attribut d'extension au nom de fichier. Les options move et rename remplacent les fichiers existants du même nom dans l'emplacement cible sans afficher de message d'avertissement ou d'erreur.

Exemple

L'exemple ci-après indique que le fichier texte source est le fichier orderStatusData.txt dans le sous-répertoire data_stream\ d'un hôte Windows et déplace le fichier traité vers le répertoire apparenté ..\done\ :

```
<?xml version="1.0" encoding="UTF-8" ?>
<FileAgent xmlns="http://cognos.obi.com/5"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:schemaLocation="http://cognos.obi.com/5 FileAgent.xsd">
<fileName>data_stream\\orderStatusData.txt</fileName>
<fileDisposal>
      <move target="..\\done\\" />
</fileDisposal>
</FileAgent>
```

Agent de système de fichiers

Un agent de système de fichiers extrait des données de table de correspondance à partir d'un fichier texte en recherchant des fichiers dans un emplacement spécifié à un intervalle défini. Lorsque l'agent localise un fichier, il extrait les données de la table de correspondance à partir du fichier source.

Tableau 15. Envoi ou extraction des agents de système de fichiers

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Non	Non	Oui

Un agent de système de fichiers comporte les attributs suivants :

Tableau 16. Attributs de l'agent de système de fichiers

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.
Chemin de fichier racine	Oui	Indique le chemin d'accès au fichier texte à utiliser par l'agent de système de fichiers.

Création d'un agent de système de fichiers

Vous pouvez créer un agent de système de fichiers afin de récupérer des données de table de correspondance à partir d'un fichier texte.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Choisissez **Système de fichiers** comme type d'agent.
5. Renseignez les zones obligatoires qui définissent les attributs de l'agent.
6. Enregistrez l'agent avec l'état activé.

Agents de file d'attente JMS (Java Messaging Service)

Un agent de file d'attente Java Messaging Service (JMS) communique avec un producteur de messages JMS via une file d'attente JMS exécutée dans l'environnement du serveur d'applications.

L'agent indique au producteur les messages JMS à envoyer. Le producteur envoie ensuite les messages concernés à la table de flux de données via l'agent.

Les agents de file d'attente JMS sont asynchrones, ce qui signifie qu'ils reçoivent des messages d'événements au fur et à mesure que les événements se produisent. En revanche, vous ne pouvez pas extraire du contexte d'un agent de file d'attente JMS.

Tableau 17. Envoi ou extraction des agents de file d'attente JMS

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Oui	Non	Non

Les attributs d'un agent de file d'attente JMS sont les suivants :

Tableau 18. Attributs des agents de file d'attente JMS

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.

Tableau 18. Attributs des agents de file d'attente JMS (suite)

Attribut	Requis	Description
Classe d'objets de connexion à la file d'attente	Oui	Identifie la classe d'objets de connexion J2EE, qui gère la file d'attente que vous voulez. Cette chaîne identifie la classe d'objets de connexion par son nom JNDI. Par exemple, jms.ManufacturingQueue. Saisissez CNowBusQueueConnectionFactoryJNDI dans cette zone. Dans un environnement Oracle WebLogic, le nom JNDI de la classe d'objets est identifié sur la console Oracle WebLogic, dans Services > JMS > Connection Factories > factoryName > General. Dans JBoss, la valeur par défaut est ConnectionFactory.
Mode Accuser réception	Non	Protocole à utiliser lorsque l'utilisateur accuse réception du message. **AUTO** : (par défaut) le fournisseur accuse réception du message lors de sa livraison. **CLIENT** : accuse réception du message lorsque l'agent le reçoit. **DUPS OK** : indique au module de publication qu'il peut envoyer un message plusieurs fois. Notez que les réceptions suivantes du même message sont considérées comme des événements nouveaux et uniques.
Nom d'utilisateur	Non	Nom d'utilisateur à utiliser pour se connecter à la classe d'objets JMS.
Mot de passe	Non	Mot de passe associé à l'utilisateur.
Propriétés JNDI	Non	(Facultatif) Propriétés Java Naming and Directory Interface (JNDI) requises pour créer l'agent ou en assurer la maintenance dans la table JMS. Ces paires nom-valeur vous permettent de spécifier les propriétés JMS reconnues par JNDI. Lorsque la file d'attente JMS s'exécute dans un espace-noms différent de celui du serveur Cognos Real-time Monitoring, vous devez définir les propriétés décrites dans «Propriétés JNDI permettant la connexion à un espace-noms distant».

Propriétés JNDI permettant la connexion à un espace-noms distant

Lorsque la file d'attente JMS s'exécute dans un espace-noms différent de celui des serveurs IBM Cognos Real-time Monitoring, définissez ces propriétés JNDI pour créer la connexion.

Les noms des propriétés JNDI sont les suivants :

```
java.naming.factory.initial
```

```
java.naming.provider.url
```

De plus, si vous utilisez la sécurité, définissez les propriétés ci-dessous :

```
java.naming.security.authentication
```

```
java.naming.security.principal
```

```
java.naming.security.credentials
```

Exemple sous Oracle WebLogic JNDI

```
java.naming.factory.initial=weblogic.jndi.WLInitiallookup
tableFactory
```

```
java.naming.provider.url=t3://localhost:9180
```

Exemple sous IBM Websphere

```
java.naming.factory.initial=com.ibm.websphere.naming.WsnInitiallookup
tableFactory
```

```
java.naming.provider.url=iiop://localhost:9180
```

Exemple sous JBOSS

```
java.naming.factory.initial=org.jnp.interfaces.Naminglookup
tableFactory
```

```
java.naming.provider.url=jnp://localhost:1099
```

Exemple sous Sun One Directory Server LDAP

```
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
```

```
java.naming.provider.url=ldap://ldap.mycompany.com:59226/dc=mycompany,dc=com
```

Enfin, appliquez le préfixe suivant à la valeur de paramètre Classe d'objets de connexion à la file d'attente : cn=

Création d'un agent de file d'attente JMS

Vous pouvez créer un agent de file d'attente Java Messaging Service (JMS) qui communique avec un producteur de messages JMS via une file d'attente JMS exécutée dans l'environnement du serveur d'applications.

Avant de commencer

Avant de créer un agent de file d'attente JMS, vous devez :

- Déployer dans le serveur d'applications hôte un bean géré par message (MDB) personnalisé, qui code en dur le nom du flux de données dans le fichier de propriétés et qui reçoit des messages de la file d'attente.

  Les serveurs d'applications n'autorisent pas les abonnements dynamiques aux files d'attente JMS. Un MDB personnalisé doit être créé à cette fin. Pour obtenir de l'aide lors de la création et la configuration de ce MDB, reportez-vous aux fichier LISEZ-MOI.txt dans le dossier des exemples sous <rép_install_c10>.
- Disposer de l'autorisation de création sur les agents. Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.
- Mentionner l'emplacement JNDI de la classe de rubriques dans le serveur d'applications qui reçoit les messages de file d'attente.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Choisissez **File d'attente JMS** comme type de source.
5. Complétez les zones requises qui définissent les attributs de l'agent.
6. Enregistrez l'agent avec l'état activé.

Agents de rubrique JMS

Un agent de rubrique JMS (Java Messaging Service) communique avec un producteur de messages JMS via une rubrique JMS exécutée dans l'environnement du serveur d'applications.

L'agent indique à l'éditeur les messages à envoyer. L'éditeur envoie ensuite les messages au flux de données via l'agent.

Les agents de rubrique JMS sont asynchrones, ce qui signifie qu'ils reçoivent des messages d'événements au fur et à mesure que les événements se produisent. En revanche, vous ne pouvez pas extraire de données de table de correspondance à partir d'un agent de rubrique JMS, comme récapitulé dans le tableau ci-dessous.

Tableau 19. Envoi ou extraction de l'agent de rubrique JMS

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Oui	Non	Non

Chaque agent JMS possède les attributs décrits dans le tableau ci-dessous :

Tableau 20. Attributs de l'agent de rubrique JMS

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents. Pour plus d'informations sur les noms d'objets, voir «Espace-noms d'objets», à la page 281.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.

Tableau 20. Attributs de l'agent de rubrique JMS (suite)

Attribut	Requis	Description
Classe d'objets de connexion à la rubrique	Oui	Identifie la classe d'objets de connexion J2EE, qui gère les rubriques que vous voulez. Cette chaîne identifie la classe d'objets de connexion par son nom JNDI, par exemple, jms.ManufacturingTopic. jms.ManufacturingQueue. Saisissez CNowBusTopicConnectionFactoryJNDI dans cette zone. Dans un environnement Oracle WebLogic, le nom JNDI de la classe d'objets est identifié sur la console Oracle WebLogic, dans Services > JMS > Connection Factories > factoryName > General. Dans JBoss, la valeur par défaut est ConnectionFactory.
Mode Accuser réception	Non	Protocole à utiliser lorsque l'utilisateur accuse réception du message. Les valeurs possibles sont les suivantes : **AUTO** : (valeur par défaut) le fournisseur accuse réception du message lors de sa livraison ; **CLIENT** : accuse réception du message lorsque l'agent le reçoit ; **DUPS OK** : Indique au module de publication qu'il peut envoyer un message plusieurs fois. Notez que les réceptions suivantes du même message sont considérées comme des événements nouveaux et uniques.
Nom d'utilisateur	Non	Nom d'utilisateur à utiliser pour se connecter à la classe d'objets JMS.
Mot de passe	Non	Mot de passe associé à l'utilisateur.
Propriétés JNDI	Non	Propriétés JNDI (Java naming and directory interface) facultatives et supplémentaires, requises pour créer l'agent et en assurer la maintenance dans la table JMS. Ces paires nom-valeur vous permettent de spécifier les propriétés JMS reconnues par JNDI. Lorsque la rubrique JMS s'exécute dans un espace-noms différent de celui des serveurs Cognos Real-time Monitoring, vous devez définir les propriétés décrites dans «Propriétés JNDI permettant la connexion à un espace-noms distant», à la page 36.

Création d'un agent de rubrique JMS

Vous pouvez créer un agent de rubrique Java Messaging Service (JMS) qui communique avec un producteur de messages JMS via une rubrique JMS exécutée dans l'environnement du serveur d'applications.

Avant de commencer

Avant de créer un agent, vous devez :

- Déployer dans le serveur d'applications hôte un bean géré par message (MDB) personnalisé, qui code en dur le nom du flux de données et qui s'abonne à la rubrique.

  Les serveurs d'applications n'autorisent pas les abonnements dynamiques aux rubriques JMS. Un MDB personnalisé doit être créé à cette fin.
- Disposer de l'autorisation de création sur les agents. Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.

- Mentionner l'emplacement JNDI de la classe de rubriques dans le serveur d'applications qui publie les rubriques.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Choisissez **Rubrique JMS** comme type de source.
5. Renseignez les zones obligatoires qui définissent les attributs de l'agent.
6. Enregistrez l'agent avec l'état activé.

Agents TIBCO Rendezvous

Un agent TIBCO Rendezvous communique avec un démon TIBCO Rendezvous s'exécutant dans l'environnement de serveur d'applications. Le démon recherche des messages dans un flux de messages TIBCO Rendezvous.

Lorsqu'il trouve un message demandé par l'une des tables TIBCO Rendezvous (voir «Tables TIBCO Rendezvous», à la page 349), il récupère ses données et les transmet à la table par le biais de l'agent.

Les agents TIBCO Rendezvous sont asynchrones, ils reçoivent les messages des événements lorsque ces événements se produisent, comme résumé dans le tableau ci-après. Vous ne pouvez pas récupérer de données de table de correspondance à partir d'un agent TIBCO Rendezvous.

Tableau 21. Envoi ou extraction des agents TIBCO Rendezvous

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Oui	Non	Non

Attributs

Cette section fournit des détails sur les attributs d'un agent TIBCO Rendezvous.

Tableau 22. Attributs de l'agent TIBCO Rendezvous

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur **Sélectionner un dossier** pour choisir un dossier.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.
Service	Non	Port du service TIBCO Rendezvous. Laissez cette zone vide pour utiliser le port par défaut 7500. Modifiez cette valeur uniquement si l'administrateur TIBCO Rendezvous vous attribue un autre port.

Tableau 22. Attributs de l'agent TIBCO Rendezvous (suite)

Attribut	Requis	Description
Réseau	Oui	Identifie l'interface réseau à utiliser si l'hôte est connecté à plusieurs réseaux ou s'il prend en charge la multidiffusion (auquel cas l'adresse ressemblera à ;222.1.2.3). Modifiez cette valeur uniquement si l'ordinateur hôte n'est pas sur le réseau par défaut, puis utilisez l'adresse IP fournie par votre administrateur TIBCO Rendezvous.
Démon	Non	Port du démon de routage sur l'hôte TIBCO Rendezvous trouvé sur le réseau et identifié par l'attribut Réseau. Laissez cette zone vide pour utiliser le port par défaut 7500. Modifiez cette valeur uniquement si l'administrateur TIBCO Rendezvous vous attribue un autre port.

Création d'un agent TIBCO Rendezvous

Cette section explique comment créer un agent qui communique avec un démon TIBCO Rendezvous s'exécutant dans l'environnement de serveur d'applications.

Avant de commencer

Avant de créer un agent, vous devez :

- Attribuer des autorisations de création aux agents. Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.
- Vous connecter au démon d'écoute TIBCO Rendezvous.

Pour vous connecter, vous avez besoin des noms Service, Réseau et Démon. Pour obtenir des valeurs particulières, contactez l'informaticien qui gère votre système TIBCO Rendezvous.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Agent**.
4. Choisissez le type de source **TIBCO Rendezvous**.
5. Renseignez les zones qui définissent les attributs de l'agent.
6. Enregistrez l'agent.

Agents de services Web

Un agent de service Web communique avec une application s'exécutant sur un serveur d'applications Web pour récupérer des données de table de correspondance. L'agent se connecte à l'application via une connexion HTTP, à l'aide du protocole SOAP (Simple Object Access Protocol).

Les agents de service sont synchrones. Par conséquent, ils récupèrent les données de table de correspondance suite à une demande spécifique.

Tableau 23. Envoi ou extraction de l'agent de service Web

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Oui	Non	Oui

Attributs des agents de services Web

Cette section fournit des détails sur les attributs d'un agent de service Web.

Tableau 24. Attributs des agents de services Web

Attribut	Requis	Description
Nom	Oui	Identifie l'agent. Ce nom doit être unique parmi les agents. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Oui	Indique si l'agent est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Oui	Dossier dans lequel enregistrer l'agent. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Description	Non	Description facultative qui peut contenir tous les caractères de texte.
Adresse URL	Oui	Emplacement HTTP du fichier WSDL qui décrit le service, les données qu'il fournit et la procédure d'échange des données avec le service. Les données renvoyées doivent être un message SOAP de style doc. La liaison RPC n'est pas prise en charge.
Nom d'utilisateur	Non	(Facultatif) Nom d'utilisateur à saisir pour se connecter au service. Ce paramètre est transmis au serveur lorsqu'il requiert un nom d'utilisateur.
Mot de passe	Non	(Facultatif) Mot de passe de l'utilisateur à saisir pour se connecter au service. Ce paramètre est transmis au serveur lorsqu'il requiert un mot de passe.

Création d'un agent de service Web

Vous pouvez créer un agent de service Web qui communique avec une application s'exécutant sur un serveur d'applications Web pour récupérer des données de table de correspondance.

Avant de commencer

Avant de créer un agent de service Web, vous devez :

- Attribuer des autorisations de création aux agents. Pour plus de détails, voir «Autorisation de création d'objets», à la page 299.
- Connaître l'emplacement HTTP du fichier WSDL qui définit le service à utiliser. Notez que le service doit publier ses données dans une liaison SOAP. La liaison RPC n'est pas prise en charge.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > > **Agent**.
4. Choisissez **Service Web** comme type d'agent.
5. Renseignez les zones qui définissent les attributs de l'agent.
6. Enregistrez l'agent avec l'état activé.

Chapitre 5. Alertes

Les alertes sont des notifications d'incidents exceptionnels envoyées à des utilisateurs ou à des systèmes externes.

Une alerte peut être un simple message indiquant qu'un incident s'est produit. Une alerte peut également être plus détaillée et contenir des informations sur la cause de l'incident ainsi que sur les solutions possibles.

Une alerte peut être autonome ou associée à une règle. Elle peut également être le clone d'une alerte existante.

Chaque message d'alerte contient du texte décrivant l'incident exceptionnel à l'abonné. Ce texte apparaît sur l'onglet **Plan de travail** et sur tous les autres appareils identifiés par le profil de diffusion de l'abonné. La présentation du message dépend de l'appareil sur lequel il est présenté à l'abonné.

Création d'une alerte autonome

Dans IBM Cognos Real-time Monitoring Workbench, vous pouvez créer une alerte autonome qui n'est associée à aucune règle.

Avant de commencer

Pour créer une alerte, vous devez disposer des autorisations suivantes :

- Autorisation de création pour des activités métier (pour plus d'informations, voir «Autorisation de création d'objets», à la page 299).
- Autorisation en lecture et écriture sur l'activité métier qui contient l'alerte.
- Autorisation en lecture seule sur la vue ou le cube qui alimente l'alerte.

Procédure

1. Dans le **Modélisateur de scénarios**, sélectionnez une activité métier existante.
2. Dans le panneau Objets du système, sélectionnez un scénario existant destiné à contenir l'alerte.
3. Cliquez sur l'onglet **Alertes**.
4. Cliquez sur **Nouvelle alerte**.
5. Choisissez la source de données qui va alimenter l'alerte.
   - Si le scénario a une vue par défaut, c'est celle-ci qui s'affiche par défaut. Pour choisir une autre source, cliquez sur **Sélectionner une source de données**.
   - S'il s'agit d'une vue, choisissez la vue.
   - S'il s'agit d'un cube, choisissez le niveau de dimension. Si vous le souhaitez, vous pouvez également appliquer un filtre limitant les données communiquées par le cube dans l'alerte.
   - Si la source contient des données, celles-ci s'affichent et vous fournissent un échantillon de ce à quoi vous pouvez vous attendre. Lorsque la source est vide, le formulaire affiche simplement les noms des colonnes et le message Aucune donnée disponible.
6. Renseignez les zones de la fenêtre Créer une alerte :
   - **Nom de l'alerte**

© Copyright IBM Corp. 2007, 2014

- **Description** (facultative)
- **Gravité**
- **Abonnés**

  Utilisez le bouton **Ajouter/Supprimer** pour sélectionner les utilisateurs et les rôles abonnés à cet événement. Il peut s'agit de **Destinataires obligatoires** ou de **Destinataires facultatifs**.
- **Objet**

  Utilisez le bouton **Ajouter une zone** pour insérer des zones dans la ligne Objet de cette alerte. Ces zones servent à insérer les valeurs réelles dans l'alerte.
- **Corps de texte** (facultatif)

  Cette zone permet de créer un message d'alerte pour les abonnés de cette alerte. Vous pouvez utiliser les boutons **Ajouter une zone**, **Ajouter un tableau récapitulatif** et **Ajouter un accusé de réception** pour rédiger un message avec des données pertinentes.

Création d'une alerte associée à une règle

Vous pouvez créer une alerte qui est associée à une règle, par opposition à une alerte autonome.

Avant de commencer

Pour créer une alerte, vous devez disposer des autorisations suivantes :

- Autorisation de création pour des activités métier (pour plus d'informations, voir «Autorisation de création d'objets», à la page 299).
- Autorisation en lecture et écriture sur l'activité métier qui contient l'alerte.
- Autorisation en lecture seule sur la vue ou le cube qui alimente l'alerte.

Procédure

1. Suivez les instructions de la section «Création de règles», à la page 321.
2. Dans l'onglet **Simple**, cliquez sur **Sélectionner**.
3. Dans la fenêtre Sélectionner l'alerte, choisissez une alerte existante ou créez-en une nouvelle en cliquant sur **Créer une alerte**.

Clonage d'une alerte existante

Vous pouvez cloner une alerte existante afin d'économiser du temps et des efforts.

Procédure

1. Copiez la définition d'une alerte existante dans une nouvelle alerte.
2. Modifiez l'alerte à cloner.
3. Changez le nom de l'alerte, ainsi que les autres attributs qui diffèrent de l'alerte d'origine.
4. Choisissez **Enregistrer sous Nouvelle alerte**.

Attributs des alertes

Il est conseillé de modifier les paramètres de notification qui contrôlent le nombre maximal d'alertes pouvant être envoyées dans un intervalle de temps spécifié.

Pour plus d'informations, voir la rubrique relative à la définition de restrictions applicables aux notifications d'alerte dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Chaque alerte possède les attributs suivants :

Nom de l'alerte
Identifie l'objet d'alerte. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être propre à chaque alerte d'un même scénario. Pour plus de détails, voir «Espace-noms d'objets», à la page 281.

Etat Spécifie si la règle est activée (réception de nouvelles informations de flux de données) ou désactivée. Si le scénario contenant l'alerte est désactivé, vous ne pouvez pas activer l'alerte. Vous devez activer le scénario pour pouvoir activer l'alerte.

Gravité
Précision du niveau d'importance d'un message. Les valeurs possibles sont ELEVEE, NORMALE (valeur par défaut) ou FAIBLE. Les messages arrivant dans Real-time Monitoring Dashboard sont classés dans le dossier correspondant à leur niveau d'importance. En outre, l'importance des courriers électroniques est signalée en conséquence par la zone d'en-tête Importance disponible en standard dans les messageries électroniques.

Description
Description facultative qui peut contenir tous les caractères de texte.

Source de données
Niveau de dimension d'une vue ou d'un cube définissant les colonnes de l'alerte. Cette source doit être identique à celle de la règle associée ou être dérivée de cette source, faute de quoi, l'alerte générée risque de contenir des informations incorrectes.

Abonnés
Utilisateurs recevant l'alerte. Pour plus de détails, voir «Abonnés aux alertes», à la page 46.

Objet Ligne de texte indiquant l'objet de l'alerte, semblable à la ligne d'objet d'un courrier électronique. Ce texte peut faire référence à des colonnes de la vue métier sous-jacente.

Corps Corps de texte de l'alerte. Ce texte peut faire référence à des colonnes de la vue métier sous-jacente et contenir des accusés de réception ainsi que des tableaux récapitulatifs. Pour plus d'informations sur les accusés de réception, voir le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*. Pour plus d'informations sur les tableaux récapitulatifs, voir Chapitre 26, «Tableaux récapitulatifs», à la page 313.

Objet et corps de texte d'un message

Chaque message d'alerte contient du texte décrivant l'incident exceptionnel à l'abonné.

Ce texte apparaît dans IBM Cognos Real-time Monitoring Workbench, IBM Cognos Real-time Monitoring Dashboard ainsi que sur tous les autres appareils identifiés par le profil de diffusion de l'abonné. La présentation du message dépend de l'appareil sur lequel il est présenté à l'abonné.

Les attributs d'alerte Objet et Corps de texte définissent le texte du message. Ces attributs contiennent du texte statique ainsi que des zones. Lorsque l'alerte génère

le message, elle remplace les zones par les valeurs des colonnes du même nom sur la ligne de la vue métier ayant entraîné l'alerte.

Prenons l'exemple de définition de message par défaut ci-après :

Objet : AVERTISSEMENT -- Un client a ouvert un ticket de problème. Corps de texte : le client CUST_NAME de niveau HIGH a ouvert un ticket de problème numéro TICKET.

Lorsque l'alerte est activée, elle génère un message semblable au courrier électronique suivant :

De : IBM Cognos Real-time Monitoring
Date : 05/03/2007 19:45
Objet : AVERTISSEMENT -- Un client a ouvert un ticket de problème

Le client Acme Works de niveau ELEVE a ouvert un ticket de problème numéro 0703.

Figure 2. Message provenant d'une alerte activée

Vous pouvez inclure un code HTML valide dans le corps du message. Si vous faites référence à un objet externe, tel qu'un graphique, référencez-le par son emplacement sur un serveur HTTP ; ne faites pas référence à un fichier local, car ce dernier ne sera pas inclus dans le message.

Abonnés aux alertes

Lorsque vous définissez une alerte, vous pouvez également déclarer un ou plusieurs utilisateurs comme destinataires de la notification d'alerte correspondante.

Grâce au bouton **Ajouter/Supprimer** en regard de la liste d'abonnés présentée par l'éditeur d'alerte, vous pouvez désigner les utilisateurs et rôles qui devront recevoir l'alerte. Vous pouvez également identifier, dans la vue de l'alerte, les colonnes présentant des listes d'utilisateurs, de rôles ou d'adresses électroniques devant recevoir la notification.

La fenêtre Avertir les abonnés comporte deux onglets : l'un pour les abonnements individuels et l'autre pour les abonnements guidés par les données.

L'onglet **Abonnement individuel** vous permet de choisir les utilisateurs qui recevront la notification.

- Les abonnés obligatoires reçoivent toujours la notification. Ils ne peuvent pas se se désabonner eux-mêmes de l'alerte dans IBM Cognos Real-time Monitoring Dashboard. Ils doivent être supprimés à partir de l'onglet **Abonnement individuel**.
- Les abonnés facultatifs reçoivent les notifications, mais peuvent se désinscrire au moyen de Cognos Real-time Monitoring Dashboard.

L'onglet **Abonnement guidé par les données** vous permet d'identifier les colonnes de la vue qui contiennent les noms ou adresses des utilisateurs ou les rôles

destinés à recevoir la notification. Une colonne de la vue peut contenir soit une liste d'utilisateurs et de rôles, soit une liste d'adresses de courrier électronique devant recevoir les notifications. Les listes de valeurs d'une colonne sont séparées les unes des autres par une virgule ou par un point-virgule, et chacun des éléments peut être éventuellement indiqué entre guillemets ("). Sélectionnez la colonne à utiliser à partir des listes déroulantes **Destinataires guidés par les données (utilisateur ou rôle)** et **Destinataires guidés par les données (adresse électronique)**.

Les abonnements guidés par les données sont obligatoires : les utilisateurs, rôles et adresses électroniques inscrits n'ont pas la possibilité de se désabonner. Lorsque les utilisateurs reçoivent ces abonnements via leur adresse électronique, ils ne les voient pas dans la liste des abonnements sur Real-time Monitoring Dashboard. Lorsque la colonne contient plusieurs instances d'une même adresse électronique, un seul message est envoyé. Toutefois, si les entrées diffèrent légèrement, un message est envoyé à chacune d'elles. Par exemple, les deux variations d'une même adresse myname@obi.com et "Name" <myname@obi.com> génèrent deux messages distincts.

Gestion des messages de notification d'alerte

Les alertes subsistent dans la liste du gestionnaire d'alertes jusqu'à ce qu'elles soient supprimées explicitement par l'utilisateur ou automatiquement par le système.

Le système supprime les messages au bout du nombre de jours spécifié par l'administrateur système dans la fenêtre Paramètres du système, comme indiqué dans les sections relatives aux paramètres système du *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Vous pouvez également contrôler le nombre maximal d'alertes pouvant être envoyées au cours d'un intervalle de temps spécifique. Pour plus d'informations, voir la rubrique relative aux restrictions applicables aux notification d'alerte dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Les utilisateurs reçoivent les notifications d'alerte sur les appareils identifiés par leur profil de diffusion dans l'onglet correspondant (voir «Onglet Profils de diffusion», à la page 356).

Pour obtenir des informations sur l'utilisation du gestionnaire d'alertes dans IBM Cognos Real-time Monitoring Dashboard pour afficher des notifications, s'y abonner ou les supprimer, voir Interaction avec les alertes dans la documentation du tableau de bord.

Etats des alertes

IBM Cognos Real-time Monitoring propose deux types d'alertes : les alertes sans état et les alertes avec état.

Les alertes sans état sont des notifications uniques signalant l'existence d'une condition métier.

Ces alertes sont envoyées lorsque la condition de règle est remplie. Par exemple, une alerte sans état peut envoyer une notification au responsable d'un entrepôt lorsque l'inventaire d'un produit tombe en deçà d'un seuil spécifique. Notez que

chaque changement ultérieur des niveaux d'inventaire de ce produit entraîne également l'envoi d'une notification tant que l'inventaire demeure en dessous de ce seuil.

Les alertes avec état présentent l'état lancé ou confirmé à la réception aussi longtemps que la condition métier existe, et l'état diminué lorsque la condition n'existe pas.

Dans le cas d'une alerte avec état, le responsable d'un entrepôt reçoit une notification d'alerte lorsque l'inventaire tombe en deçà d'un seuil défini, et n'en reçoit pas d'autre jusqu'à ce que l'alerte soit diminuée, vraisemblablement lorsque les niveaux d'inventaire remontent au-dessus du seuil. Lorsque plusieurs intervenants sont concernés par une alerte, l'un d'eux peut choisir de gérer l'alerte lancée et d'en accuser réception. Il effectue cette opération en cliquant sur le lien Accuser réception situé dans le corps du message. Le créateur de l'alerte insère ce lien dans le corps du message au moment où il crée l'alerte. Pour plus d'informations, voir la rubrique relative aux accusés de réception dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Figure 3. Etats des alertes

En combinant une alerte avec état et une période Correspondant à, vous pouvez différer la notification. Par exemple, vous pouvez choisir de n'alerter le responsable d'un entrepôt que lorsque l'inventaire d'un produit est resté en deçà d'un seuil défini pendant une journée entière : dans ce cas, la condition d'alerte *correspond à* 1 jour. De cette façon, le responsable ne reçoit aucune notification si l'inventaire tombe en dessous du seuil spécifié la veille même d'un réapprovisionnement.

Escalade d'alerte

Vous pouvez contrôler l'état d'une alerte et générer de nouvelles alertes lorsque les conditions l'exigent.

Par exemple, si une alerte n'a pas été traitée dans les délais, une nouvelle alerte peut être envoyée à des utilisateurs occupant un poste plus élevé dans la hiérarchie, entraînant ainsi l'escalade de l'alerte d'origine. Pour tester ces états, utilisez la fonction IS_RAISED («IS_RAISED», à la page 165). Pour plus de détails, voir «Règles contrôlant les alertes», à la page 325.

Consolidation de plusieurs messages

Les événements contiennent parfois plusieurs lignes d'information. Lorsqu'un événement remplit une condition de règle, cette règle génère une alerte pour chacune des lignes de l'événement.

Il est généralement préférable de n'envoyer qu'un seul message décrivant la totalité des alertes. Ce type de message est appelé alerte consolidée.

Prenons l'exemple de l'introduction d'un nouveau bon de commande dans le système (événement). Si la quantité d'articles en stock est insuffisante pour le traitement de cette commande, il est possible de signaler cette condition à l'aide d'une alerte. Lorsque plusieurs articles de la commande n'existent pas en quantité suffisante dans le stock, chacun d'eux génère une nouvelle alerte. Si vous préférez n'envoyer qu'une seule notification à la place, utilisez une alerte consolidée.

Procédure

Dans la fenêtre Créer une règle, sélectionnez l'option **Consolider plusieurs messages à partir d'un même événement**.

Configuration d'une alerte pour l'appel d'un service Web externe

Pour configurer un message d'alerte permettant d'appeler un service web externe, vous devez concevoir une fonction Java créant le message du service web à partir de la charge utile de l'alerte.

Cette fonction détermine les éléments de données de la charge utile de l'alerte qui sont mis en correspondance avec les entrées attendues par le service web.

Cette fonction est ensuite téléchargée dans IBM Cognos Real-time Monitoring sous la forme d'une fonction définie par l'utilisateur (UDF). Le service web est spécifié dans un profil de service web qui appelle l'UDF. Vous pouvez ensuite définir le profil de service web en tant qu'abonné à l'alerte de votre choix.

La création de cette fonction requiert une grande connaissance de la programmation. Pour en savoir davantage, voir «Fonction définie par l'utilisateur associée au service Web», à la page 50.

Procédure

1. Créez une nouvelle alerte.

   N'abonnez aucun utilisateur à cette alerte pour le moment. Vous définirez des utilisateurs ultérieurement.
2. Créez une fonction chargée de concevoir le message du service web à l'aide de certaines parties des données d'alerte et de les regrouper sous la forme d'un fichier JAR.

   Pour en savoir davantage, voir «Fonction définie par l'utilisateur associée au service Web», à la page 50.
3. Téléchargez le fichier JAR dans le plan de travail sous la forme d'une fonction UDF.

   Pour en savoir davantage, voir «Création et utilisation d'une fonction définie par l'utilisateur», à la page 360.
4. Créez un utilisateur, puis enregistrez-le.

   Vous devez créer et enregistrer l'utilisateur pour pouvoir modifier les propriétés du profil de tableau de bord par défaut.
5. Modifiez le paramétrage du nouvel utilisateur en procédant comme suit :
   - Dans la fenêtre Modifier l'utilisateur, cliquez sur l'onglet **Profils de diffusion**.
   - Ouvrez le profil de tableau de bord par défaut.
   - Désélectionnez l'option **Ajouter automatiquement ce profil...**.

- Cliquez sur **OK**.

6. Créez un profil de diffusion pour le nouvel utilisateur en procédant comme suit :
   - Dans la fenêtre Modifier l'utilisateur, cliquez sur l'onglet **Profils de diffusion**.
   - Cliquez sur l'option **Créer un profil**.
   - Dans la zone **Type de profil** de la fenêtre qui s'affiche, sélectionnez **Service Web**.
   - Dans la zone **Nom du profil**, entrez une valeur appropriée.
   - Dans la zone **Adresse URL du service Web**, entrez le point limite du WSDL.
   - Dans la zone **Méthode**, spécifiez la méthode (opération) que vous souhaitez appeler dans le service web.
   - Renseignez les zones **Nom d'utilisateur** et **Mot de passe**, en fonction des besoins.
   - Dans la zone **UDF**, sélectionnez le fichier JAR que vous avez téléchargé pour l'appel de ce service web.
   - Assurez-vous que l'option **Ajouter automatiquement ce profil...** est désélectionnée.
   - Cliquez sur **OK**.
7. Réaffichez la configuration de l'alerte que vous avez créée à l'étape 1.
8. Ajoutez en tant qu'abonné l'utilisateur que vous venez de créer.
9. Cliquez sur **Enregistrer**.

   L'envoi de l'alerte entraînera l'appel de ce service web.

Fonction définie par l'utilisateur associée au service Web

Cette fonction reçoit en entrée un élément DOM décrivant les données d'alerte générées par IBM Cognos Real-time Monitoring et conforme au fichier alertMessage.xsd. Elle génère en sortie un message AXIS contenant le protocole SOAP envoyé au service Web.

La fonction que vous créez doit étendre la classe `com.cognos.obi.api.function.webservice. IAlertWSMessageConstructor`.

Interface com.cognos.obi.api.function.webservice.IAlertWSMessageConstructor :

```
package com.cognos.obi.api.function.webservice;
import com.cognos.obi.api.function.*;
import org.apache.axis.Message;
import org.w3c.dom.Element;
import java.lang.Exception;
public interface IAlertWSMessageConstructor extends IUDFunction
{
public Message constructMessage(Element alertData, String
wsdlURI,
String soapOperation) throws Exception;
}
```

Ajout de tableaux récapitulatifs dans le corps de la notification d'alerte

Les tableaux récapitulatifs décrivent le contenu d'une vue et présentent ces informations dans un rapport joint au message d'alerte.

Les tableaux récapitulatifs présentent souvent des informations qui replacent l'événement dans son contexte.

Les alertes affichées dans le tableau de bord opérationnel incorporent le tableau récapitulatif sous la forme d'une table HTML en ligne. L'option **Envoyer en tant que** spécifie le format du tableau récapitulatif à joindre à la notification d'alerte envoyée aux profils d'abonnement des utilisateurs. Vous pouvez incorporer le tableau récapitulatif dans le corps du message ou l'inclure en tant que pièce jointe dans l'un des formats disponibles.

Procédure

1. Ouvrez la fenêtre Ajouter un tableau récapitulatif.
2. Sélectionnez le tableau récapitulatif dans la liste et cliquez sur **OK** pour l'ajouter à l'alerte.

Filtrage des tableaux récapitulatifs

Lorsque vous créez le tableau récapitulatif, vous identifiez la vue ou le cube dont les données alimentent le tableau. Toutefois, les données qui s'affichent dans le tableau récapitulatif dépendent du type de la source (sans état ou avec état) et du filtrage des lignes de données.

Le filtrage des lignes de données est spécifié dans la boîte de dialogue **Ajouter un tableau récapitulatif**. En règle générale :

- L'option **Données du tableau récapitulatif basées sur** indique si la source avec état d'un tableau récapitulatif doit inclure les données reposant sur tous les événements ou uniquement sur ceux qui ont rempli la condition de règle.
- L'option **Les données du tableau récapitulatif sont** indique si les lignes de la source du tableau récapitulatif doivent n'inclure que celles associées à l'événement ayant activé l'alerte, ou également toutes les données d'événements précédentes de la source.

Pour mieux comprendre l'effet de ces paramètres sur les données du tableau récapitulatif, examinez les deux vues ci-après qui suivent et présentent les commandes de produits. La vue sans état OrderDetails est un résumé de chaque événement de commande, tandis que la vue avec état OrderAggregates assure le suivi de la quantité moyenne commandée pour chaque produit.

Examinez à présent les événements suivants :

```
prod_name  ord_qty
```

```
---------  -------
nails         1000
plywood       1000
nails         4000
nails         4000
plywood       5000
```

Une fois les événements entrés dans le flux de données, la vue *OrderAggregates* comporte les valeurs suivantes :

```
prod_name  AVG(ord_qty) Ct
---------  -----------  --
nails            3000   3
plywood          3000   2
```

Par défaut, un rapport récapitulatif utilisant la vue OrderAggregates présente les détails des deux produits, quel que soit le produit associé à l'événement qui a généré l'alerte. En outre, même si les trois derniers événements ont rempli la condition de règle ord_qty supérieur à 3 000, le tableau récapitulatif affiche les résultats de tous les événements, ce qui ne correspond pas obligatoirement à ce que vous vouliez obtenir.

Les deux options de filtrage de la boîte de dialogue **Ajouter un tableau récapitulatif** modifient les résultats en filtrant ceux qui sont présentés dans le tableau.

L'activation de l'option **Les données du tableau récapitulatif sont** entraîne l'affichage de éléments suivants dans le tableau récapitulatif :

- toutes les données de la vue ou de la face du cube ;
- uniquement les données liées à l'événement identifié par la règle.

Pour n'afficher que les données liées à l'événement, vous devez définir la relation existant entre la source de l'événement et celle du tableau récapitulatif. Par exemple, si vous souhaitez que le tableau récapitulatif n'affiche que le résultat de l'élément "product" dans les deux vues, définissez cette relation en sélectionnant la colonne prod_name dans les deux vues. Vous indiquez ainsi au tableau récapitulatif de ne présenter que les lignes de la vue OrderAggregates dont la valeur prod_name correspond au nom de produit dans la vue OrderDetails. En conséquence, le tableau récapitulatif n'affichera la valeur du produit "nails" que lorsque la règle générera l'alerte.

```
prod_name  AVG(ord_qty) Ct
---------  -----------  --
nails            3000   3
```

De la même façon, si vous travaillez avec la face d'un cube, vous devez sélectionner les colonnes identifiant le mieux l'événement pour le tableau récapitulatif.

Option Données du tableau récapitulatif basées sur

Cette option indique si une vue avec état doit inclure les données reposant sur tous les événements ou uniquement sur ceux qui ont rempli la condition de règle.

Selon l'exemple décrit dans «Filtrage des tableaux récapitulatifs», à la page 51, la colonne AVG(ord_qty) de la vue OrderAggregates présente une valeur de 3 000

pour le produit "nails" après le traitement des deux événements. Vous obtenez ce résultat lorsque cette option est paramétrée sur **Données de flux de données**. Toutefois, la condition de règle indique de ne générer une alerte que lorsque la quantité de la commande est supérieure à 3 000. Pour ne suivre que les événements ayant rempli cette condition, reparamétrez cette option sur **Filtre de règles de données de flux de données**. Le tableau récapitulatif vous présente alors une valeur moyenne de 4 000, car cette valeur correspond à la moyenne des deux événements supérieurs à 3 000.

Les illustrations ci-après présentent les résultats des vues correspondant aux données d'exemple lorsque vous utilisez les deux options de filtrage du tableau récapitulatif. Notez que le premier événement ne répond pas aux critères du filtre de règle et n'apparaît donc pas dans ces vues.

Figure 4. Résultats des vues correspondant aux données d'exemple

Le deuxième événement ne répond pas non plus aux critères du filtre de règle. Notez cependant que la vue qui présente les données liées à l'événement ne contient plus désormais que l'événement "plywood".

Figure 5. Résultats des vues correspondant aux données d'exemple

Le troisième événement répond aux critères du filtre de règle et apparaît donc dans les vues ci-après. Une fois de plus, le produit "nails" est le produit apparaissant dans les vues liées à l'événement.

Evénements

prod_name	ord_qty	Ct
nails	1000	
plywood	1000	
nails	**4000**	

	Toutes les données de la vue tableau récapitulatif			Uniquement les données liées à l'événement (prod_name)		
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Données de flux de données	nails	2500	2	nails	2500	2
	plywood	1000	1			
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Filtre de règle de données de flux de données (ord_qty>3000)	nails	4000	1	nails	4000	1

Figure 6. Résultats des vues correspondant aux données d'exemple

Le quatrième événement lié au produit "nails" répond lui aussi aux critères du filtre de règle.

Evénements

prod_name	ord_qty	Ct
nails	1000	
plywood	1000	
nails	4000	
nails	**4000**	

	Toutes les données de la vue tableau récapitulatif			Uniquement les données liées à l'événement (prod_name)		
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Données de flux de données	nails	3000	3	plywood	3000	
	plywood	1000	1			
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Filtre de règle de données de flux de données (ord_qty>3000)	nails	4000	2	nails	4000	2

Figure 7. Résultats des vues correspondant aux données d'exemple

Le dernier événement modifie de nouveau les vues liées à l'événement. Notez que la vue présente tous les événements, tandis que la vue du filtre de règle ne contient que les événements répondant aux critères du filtre de règle.

Evénements

prod_name	ord_qty	Ct
nails	1000	
plywood	1000	
nails	4000	
nails	4000	
plywood	**5000**	

	Toutes les données de la vue tableau récapitulatif			Uniquement les données liées à l'événement (prod_name)		
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Données de flux de données	nails	3000	3	plywood	3000	2
	plywood	3000	2			
	prod_name	ord_qty	Ct	prod_name	ord_qty	Ct
Filtre de règle de données de flux de données (ord_qty>3000)	nails	4000	2	plywood	5000	1
	plywood	5000	1			

Chapitre 6. Règles métier

Le présent chapitre décrit les règles métier régissant le suivi des flux de données et la production des messages décrivant les conditions d'alerte.

A propos des règles métier

Les règles métier contrôlent les flux de données à la recherche de conditions commerciales exceptionnelles et génèrent des messages d'alerte qui décrivent ces conditions. Les règles peuvent également contrôler une condition détectée et identifier leur expiration.

Dans IBM Cognos Real-time Monitoring Dashboard, vous pouvez créer des règles métier à partir de modèles qui prédéfinissent la logique conditionnelle. Vous devez alors choisir la condition de la règle et identifier les valeurs à tester. Le système vous avertit dès que la condition est remplie. Par exemple, avec la règle "Lorsque le total de la commande est supérieur à *montant*", vous identifiez la valeur du montant, puis vous recevez une notification à chaque fois qu'une commande est supérieure au montant choisi.

Cette section explique comment créer des règles métier à partir de modèles et comment les gérer dans Cognos Real-time Monitoring Dashboard. Les modèles de règle permettent à l'utilisateur expérimenté de créer une condition de règle complexe et de définir des paramètres pour certaines valeurs de sorte que les utilisateurs de Cognos Real-time Monitoring Dashboard puissent créer des instances personnalisées du modèle. Par exemple, un utilisateur souhaite instancier une règle métier dans laquelle le montant des ventes est supérieur à 20 000 $, alors qu'un autre utilisateur souhaite être averti pour toute commande supérieure à 5 000 $. Les modèles de règle métier sont créés dans Cognos Real-time Monitoring Workbench. Pour plus d'informations, voir le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Avertissement : IBM Cognos Real-time Monitoring prend en charge un autre type de règle métier : les règles définies et entièrement gérées par le **Modélisateur de scénarios** dans IBM Cognos Real-time Monitoring Workbench. Pour en savoir davantage, reportez-vous à la documentation technique du plan de travail.

Le **Gestionnaire de règles métier** répertorie les règles auxquelles vous avez accès et vous permet de les créer, les gérer et les supprimer.

La suppression d'une règle métier n'a pas d'impact sur les messages d'alerte du **Gestionnaire d'alertes**. Tous les messages générés à partir d'une règle métier sont conservés par le **Gestionnaire d'alertes** jusqu'à leur retrait.

Création d'une règle métier

Les règles métier sont créées à l'aide du Gestionnaire de règles métier.

Vous devez au minimum disposer de l'accès en **lecture** sur l'activité métier du **Modélisateur de scénarios** dans lequel le modèle de règle est défini.

© Copyright IBM Corp. 2007, 2014

Pour plus d'informations, voir la rubrique relative aux droits dans le *Guide de référence de la modélisation d'IBM Cognos Real-time Monitoring Workbench* et la rubrique sur l'utilisation des modèles de règle dans le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Procédure

1. Ouvrez le **Gestionnaire de règles métier** et cliquez sur **Créer une règle métier**.
2. Sélectionnez le modèle de règle à utiliser.
3. Identifiez les valeurs de paramètre de la règle.
4. Donnez un nom et, en option, une description à la règle.
5. Enregistrez la nouvelle règle.

   La nouvelle règle s'affiche dans le **Gestionnaire de règles métier** et commence immédiatement à rechercher de nouvelles conditions exceptionnelles.

   Avertissement : Les règles métier sont basées sur la définition du modèle de règle tel qu'il existait au moment de leur création. Toute modification ultérieure du modèle (comprenant les messages d'alerte et les tableaux récapitulatifs) n'affecte pas les règles métier existantes créées à partir de ce modèle.

Chapitre 7. Infrastructure d'événements communs

L'infrastructure d'événements communs (CEI) est un service capable de créer, stocker et distribuer des événements.

IBM WebSphere s'appuie sur l'infrastructure d'événements communs. Cette dernière reçoit les informations système ou commerciales encapsulées dans un événement CBE. Lorsqu'un événement important se produit, un événement CBE est créé et envoyé au serveur CEI à l'aide d'un émetteur sur lequel il peut être conservé et/ou distribué.

IBM Cognos Real-time Monitoring offre la possibilité de communiquer avec un serveur qui utilise l'infrastructure d'événements communs.

Fonctionnement de l'infrastructure d'événements communs

IBM Cognos Real-time Monitoring communique via un servlet et un bean géré par message (MDB).

Le bean géré par message est déployé sur le serveur exécutant l'infrastructure d'événements communs (CEI). Il s'abonne à la file d'attente ou à la rubrique JMS qui est référencée par le processus de transmission JMS. Ce processus est à son tour référencé par l'émetteur de l'infrastructure d'événements communs. Tous les messages sont sérialisés puis envoyés à Cognos Real-time Monitoring via HTTP à l'aide d'un servlet Post HTTP. La requête HTTP provenant du bean géré par message inclut des paramètres qui dirigent le message vers le flux de données CEI dans Cognos Real-time Monitoring.

Le diagramme ci-dessous illustre la façon dont un message CEI est distribué à Cognos Real-time Monitoring.

Flux de données Infrastructure d'événements communs

Les CBE sont des documents XML et le flux de données Infrastructure d'événements communs (CEI) les traite en tant que tels.

Vous pouvez extraire des lignes et des colonnes à l'aide d'expressions XPath comme vous le feriez pour des fichiers à plat XML. Le flux de données CEI traite

© Copyright IBM Corp. 2007, 2014

l'événement CBE de la même manière qu'un fichier à plat. Par conséquent, vous devez configurer vos événements de la même façon que vous configureriez les événements d'un fichier à plat.

Prérequis à la création d'un flux de données CEI

Avant de créer un flux de données CEI, vous devez effectuer un certain nombre de tâches.

Ces tâches sont les suivantes :

- Créer un bean géré par message (MDB) CEI.
- Déployer le MDB CEI.
- Rechercher un exemple de fichier à plat (facultatif).
- Configurer l'infrastructure CEI de manière qu'elle se connecte au MDB.

IBM WebSphere Process Server doit être installé, configuré et en cours d'exécution pour CEI.

Un bean géré par message CEI est disponible dans le répertoire `webcontent/samples/sdk/cei`.

Pour que la création du fichier `cei_mdb.ear` aboutisse lors de l'exécution du test de l'infrastructure CEI, vous devez disposer des fichiers JAR suivants :

- `emf.jar`
- `com.ibm.events.client_6.1.0.jar`

Pour que la création du fichier `eventsimulator.war` aboutisse lors de l'exécution du test de l'infrastructure CEI, vous devez disposer des fichiers JAR suivants :

- `ws_runtime.jar`
- `com.ibm.events.client_6.1.0.jar`

Pour le fichier `ws_runtime.jar`, à partir de l'emplacement d'installation d'IBM WebSphere, accédez à *`emplacement d'installation`*`/usr/IBM/WebSphere/AppServer/deploytool/itp/plugins/com.ibm.websphere.v61_6.1.200`.

Pour le fichier `com.ibm.events.client_6.1.0.jar`, à partir de l'emplacement d'installation d'IBM WebSphere, accédez à *`emplacement d'installation`*`/usr/IBM/WebSphere/AppServer/plugins`.

Le tableau ci-dessous décrit les attributs d'un flux de données CEI.

Tableau 25. Attributs des flux de données CEI

Attribut	Description
Nom	Identifie la table. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).

Tableau 25. Attributs des flux de données CEI (suite)

Attribut	Description
Description	Description facultative qui peut contenir tous les caractères de texte.
Enregistrer dans	Indique le dossier dans lequel enregistrer l'agent. La valeur par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Informations sur la colonne	Les zones **Informations sur la colonne** déterminent comment placer les zones du message CEI dans les colonnes de la table de flux de données. Chaque zone est associée à une colonne dans la table de flux de données. Pour en savoir davantage, voir «Informations sur la colonne Infrastructure d'événements communs», à la page 60.
Intervalle de suppression de l'état	Cet onglet contient plusieurs options permettant de supprimer les données d'événements sauvegardées. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.

Déploiement d'un bean géré par message

Vous pouvez déployer un bean géré par message.

Procédure

1. Connectez-vous à la console Process Server Integrated Solutions.
2. Accédez à Applications, Enterprise, Installer.
3. Recherchez le fichier `cei_mdb.ear` dans le dossier déployé et téléchargez-le.

Configuration de l'infrastructure CEI (Common Event Infrastructure) pour la connexion au bean géré par message

Vous pouvez configurer l'infrastructure CEI afin qu'elle se connecte au MDB.

Procédure

1. Assurez-vous que la configuration de votre application CEI dispose des autorisations nécessaires aux émetteurs.
2. Accédez à la console d'administration d'IBM WebSphere.
3. Vérifiez que la rubrique ou la file d'attente que vous avez définie dans votre MDB existe. Celle-ci doit correspondre à QueueConnectionFactory.
4. Vérifiez que vous disposez d'une **Spécification d'activation** qui va associer la rubrique ou la file d'attente au MDB. Lorsque vous déployez le MDB, cette option vous permet de vous abonner à la rubrique ou à la file d'attente.
5. Configurez la **transmission JMS** de manière à spécifier la rubrique ou la file d'attente à laquelle le MDB est abonné, et attribuez un nom JNDI à cette transmission.

6. Configurez l'émetteur CEI utilisé par l'application de l'infrastructure de manière que l'émetteur fasse référence à la **transmission JMS** par le biais du nom JNDI que vous avez attribué à l'étape 5.

Création d'une table de flux de données à partir d'une connexion d'infrastructure d'événements communs

Une table de flux de données peut être créée à partir d'une connexion d'infrastructure d'événements communs (CEI).

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer nouveau**.
4. Cliquez sur **Flux de données**.
5. Sélectionnez **Flux de données simple**, puis cliquez sur **Continuer**
6. Sélectionnez **Infrastructure d'événements communs**, puis cliquez sur **Continuer**.
7. Sous **Informations sur la colonne**, définissez les colonnes de la table de flux de données.
8. Enregistrez la table de données CEI avec l'état activé.

Informations sur la colonne Infrastructure d'événements communs

Les zones Informations sur la colonne déterminent comment placer les zones du message JMS dans les colonnes de la table de flux de données. Cette dernière contient une colonne pour chaque zone.

Chaque zone du message peut correspondre à une zone simple mappée directement à une colonne de flux de données, ou à une zone plus complexe (zone de fichier à plat) qui comporte plusieurs zones mappées individuellement à des colonnes de la table. Les zones complexes sont considérées comme des fichiers à plat aux formats délimité (CSV), largeur fixe ou XML. Pour obtenir une description détaillée de ces types de fichiers, voir Chapitre 13, «Fichiers à plat», à la page 103.

Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-dessous.

Tableau 26. Attributs des colonnes CEI

Attribut	Description
Nom de zone	Nom de la colonne dans la table de flux de données.
Type de données	Type de données de la colonne de flux de données. Pour plus d'informations sur les types de données, voir «Types de données JMS (Java Messaging Service)», à la page 224.
Format	(Facultatif) Format de la colonne de flux de données pour les valeurs VARCHAR (chaîne) et DECIMAL.

Vous ajoutez des colonnes en cliquant sur l'option **Ajouter une zone** ou **Ajouter une zone Fichier à plat**.

Intervalle de suppression de l'état

Plusieurs options sont disponibles pour supprimer les données en mémoires de toutes les vues et les cubes qui reçoivent des données d'un flux de données.

L'onglet **Intervalle de suppression de l'état** comporte les options suivantes :

- **Ne pas supprimer l'état**

  Il s'agit de la valeur par défaut. Les données en mémoire ne sont pas effacées et elles sont propagées à partir du flux de données vers les vues, les objets et les tableaux de bord.
- **Suppression de l'état sur un calendrier**

  Lorsque vous la sélectionnez, les commandes de planification **Modifier la source de durée**, **Ajouter un calendrier**, **Modifier le calendrier** et **Supprimer le calendrier** sont activées.
- **Toujours supprimer l'état pour les nouvelles données**

  Les données en mémoire de chaque vue et cube sont effacées avant le chargement de nouvelles lignes de données dans les vues ou les cubes. Chaque ligne de données appartenant à une même interrogation peut déclencher l'effacement de l'état en mémoire des données. Sélectionnez **Traiter toutes les lignes de l'ensemble de résultats comme un seul événement** pour conserver plusieurs lignes en mémoire.

Suppression de l'état sur un calendrier

Vous pouvez utiliser cette option pour activer les commandes de planification.

L'option **Ajouter un calendrier** vous permet d'effectuer les actions suivantes :

- Supprimer l'état à une heure précise ou au cours d'un intervalle de temps donné.
- Supprimer l'état quotidiennement, un jour spécifique de la semaine ou un ou plusieurs jours spécifiques du mois.
- Supprimer l'état au cours d'un ou de plusieurs mois spécifiques.

Par exemple, pour planifier l'intervalle de suppression de l'état trimestriellement le 30 du mois à 22 heures :

1. Sélectionnez **Ajouter une heure précise** et définissez **Programmer à** sur 22 heures.
2. Sélectionnez **Chacun des jours du mois sélectionnés ci-dessous** et choisissez **30**.
3. Sélectionnez les mois de mars, juin, septembre et décembre.

Cliquez sur **Modifier le calendrier** pour éditer le calendrier sélectionné.

Cliquez sur **Supprimer le calendrier** pour supprimer le calendrier sélectionné.

Modification de la source de durée

Vous pouvez choisir les options de temps qui indiquent le moment auquel les données doivent être supprimées.

Modifier la source de durée ouvre une fenêtre dans laquelle vous pouvez sélectionner le moment auquel les données doivent être effacées, en fonction des options suivantes :

- **A l'heure indiquée**

Les données sont effacées uniquement à l'heure indiquée sur le calendrier. Lorsque vous sélectionnez cette option, il se peut qu'aucune donnée ne s'affiche dans la vue pendant un certain laps de temps. Pour éviter ce problème, sélectionnez l'option **Lorsque les données arrivent après l'heure indiquée**.

- **Lorsque les données arrivent après l'heure indiquée**

  Les données existantes sont effacées au fur et à mesure que de nouvelles données arrivent après l'heure indiquée dans le calendrier.

- **Lorsque la valeur dans la colonne d'horodatage est antérieure à l'heure indiquée**

  Les données sont effacées en fonction de la valeur définie dans la colonne d'horodatage. Si cette colonne comporte une heure antérieure à celle indiquée dans le calendrier, les données existantes sont effacées. En revanche, si l'heure indiquée dans la colonne d'horodatage est ultérieure à celle définie dans le calendrier, aucune donnée n'est effacée. La liste déroulante **Sélectionner une colonne** permet de choisir la colonne à utiliser pour les horodatages. Si aucune colonne d'horodatage n'existe, cette option n'est pas disponible.

Chapitre 8. Cubes

Un cube est un ensemble de données organisées par dimensions et mesures et ce, dans le but de rassembler différents sous-ensembles d'un plus grand ensemble de données.

Lorsqu'ils prennent la forme d'un objet de tableau de bord, les cubes vous permettent de sélectionner rapidement des catégories qui filtrent les données afin d'afficher des résultats conformes à votre sélection. Par exemple, un cube de données de ventes peut fournir des agrégations des mêmes données par produit, heure ou par dimension de région de vente. En regardant le cube, vous pouvez choisir d'afficher le total des ventes d'un produit Nails (Clous) pour une région donnée West (Ouest) au cours d'un trimestre fiscal Q1 (T1) :

```
West
   Q1   January   Hardware
 Nails   120,000
West
   Q1
 March     Hardware   Nails
  98,000
Total                                        218,000
```

En outre, en supprimant rapidement la spécification de la dimension Product, vous demandez à voir la totalité des ventes pour cette région et ce trimestre :

```
West
   Q1
 January   Hardware   Nails
 120,000
West
   Q1
 March     Hardware   Nails
  98,000
West
   Q1
 March     Hardware   Screws
 97,000
West
   Q1
 January   Lumber     Studs
 137,000
Total                                        452,000
```

Ou la totalité des ventes au cours du mois de mars :

```
West   Q1
 March     Hardware   Nails
  98,000
West   Q1
 March     Hardware   Screws
 97,000
East   Q1
 March     Lumber     Plywood
92,000
South  Q1
 March     Hardware   Nails
  98,000
```

© Copyright IBM Corp. 2007, 2014

```
Total                                        385,000
```

Ou encore la totalité des ventes de la gamme de produits Lumber pour la région West (Ouest) :

```
West   Q1
January    Lumber     Studs
 137,000
Total                                        137,000
```

Pour plus d'informations, consultez des exemples de graphiques et de tables de cubes dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Dashboard.*

Mesures

Les mesures correspondent aux valeurs centrales agrégées et analysées.

Dans les exemples ci-dessus, total sales correspond à la valeur agrégée. Dans chacun de ces exemples, la valeur "total" mesure la somme de toutes les ventes de l'ensemble. Les mesures sont effectuées à l'aide des fonctions d'ensembles (Set) C-SQL, telles que SUM, AVG, MIN, MAX, STD_DEVIATION et VARIANCE.

Pour en savoir plus sur les mesures, voir «Colonnes Mesure», à la page 66.

Dimensions dans les cubes

Une dimension est un ordre de classement qui, pour chacun des niveaux du plus élevé au plus bas, décrit des ensembles de données associées plus petits et plus distincts.

Dans les exemples fournis dans la rubrique relative aux cubes, la région d'activité constitue un niveau d'une dimension géographique, les colonnes trimestre et mois correspondent à chacun des niveaux d'une dimension de temps, et les gammes et noms de produit font partie d'une dimension d'inventaire. Dans la dimension de temps, les mois constituent des sous-ensembles des trimestres fiscaux, de même que le nom de produit correspond à un sous-ensemble du niveau Gamme de produit. Le tableau ci-après présente quelques exemples de dimensions :

Tableau 27. Exemples de dimensions

Heure	Géographie	Inventaire	Sécurité	Taxinomie
année	continent	classification	type	règne
trimestre	pays	type	classement	embranchement
mois	région	fabricant	société	classe
semaine	département	modèle	CUSIP	ordre
jour	comté	configuration		famille
heure	ville			genre
minute	quartier			espèce

Pour en savoir davantage, voir Chapitre 11, «Dimensions», à la page 91.

Vous pouvez limiter l'accès des utilisateurs aux données du cube à l'aide d'un filtre d'accès. Pour en savoir davantage, voir Chapitre 2, «Filtres d'accès», à la page 3.

Création de cubes

Les cubes sont semblables aux vues métier, dans le sens où ils rassemblent des données d'événements, mais ils le font sur plusieurs dimensions.

La vue qu'un cube agrège est une table de faits : une vue ou table à l'intérieur d'un événement qui contient une ou plusieurs colonnes à mesurer (agréger) et qui contient aussi des colonnes qui identifient les éléments dimensionnels associés à l'événement. Vous pouvez par exemple imaginer une table de faits contenant un événement semblable à celui-ci :

```
Cost   Quantity Product State      Month
------ -------- ------- ---------- -------
200.00     1600 Nails   California January
```

Toutefois, en pratique, les éléments dimensionnels sont stockés dans des dimensions (tables de correspondance spéciales) et référencés par ID, comme suit :

```
Cost   Quantity prod_id region_id ddim_id
------ -------- ------- --------- -------
200.00     1600     100         7      39
```

Cette illustration montre un cube conçu à partir de la table de faits OrderDetails (Détails de la commande) qui mesure le total des ventes sur plusieurs régions, produits et durées.

Figure 8. Cube construit à partir de la table de faits OrderDetails

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez **Cubes**.
3. Dans le menu **Activités**, cliquez sur **Créer**.
4. Cliquez sur **Parcourir** et choisissez une table de faits ou des tables et des colonnes à mesurer.
5. Indiquez la façon de les mesurer (formules agrégées à utiliser).

6. Choisissez une ou plusieurs dimensions classant les mesures.

Colonnes Mesure

Les colonnes Mesure définissent les agrégations calculées par le cube. Un cube doit disposer d'au moins une colonne Mesure. Chaque colonne Mesure définit une expression qui contient une fonction d'ensembles (Set) C-SQL agrégeant d'autres colonnes de la table de faits.

Ainsi, pour déterminer le "total des ventes" de la table de faits OrderDetails (Détails de la commande), une colonne Mesure sera définie de la sorte :

```
SUM(OrderDetails.prod_cost*OrderDetails.order_qty) AS
TotalOrderSales
```

Création d'un cube avec des colonnes de dimension

Les colonnes Dimension classent les mesures. Un cube doit disposer d'au moins une dimension. De plus, les données présentes dans la table de faits doivent pouvoir identifier un élément unique dans chaque dimension associée.

Pour en savoir davantage, voir Chapitre 11, «Dimensions», à la page 91.

Avant de commencer

Avant de créer un cube, vous devez disposer d'une autorisation de création sur les vues, les cubes et les dimensions. Vous devez également disposer des éléments suivants :

- Au moins d'un accès en lecture seule à la dimension à inclure. Pour plus de détails, voir «Création de dimensions», à la page 94.
- Une dimension avec l'option **Catégories géographiques** activée si vous prévoyez d'utiliser le cube pour des graphiques géographiques.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Cube**.
4. Attribuez un nom au cube dans la zone **Nom** et indiquez, éventuellement, une description du cube.
5. Choisissez la **Table Fait** contenant les données à mesurer ainsi que les colonnes permettant d'identifier les éléments de dimension. Cliquez sur **Parcourir** pour sélectionner la **Table fait**.
6. Dans la boîte de dialogue **Sélectionner un objet**, choisissez l'objet à utiliser avec la **Table Fait** et cliquez sur le bouton **OK**.
7. Définissez une ou plusieurs «Colonnes Mesure».
   - Cliquez sur **Ajouter une colonne Mesure** pour définir une colonne.
   - Nommez la colonne dans la zone **Nom de la mesure**.
   - Définissez la formule de la mesure par le biais d'une «Fonctions d'ensembles», à la page 126 C-SQL dans la zone **Expression agrégée**. La fonction doit se rapporter à une colonne de la table de faits. Par exemple, l'expression SUM() ci-après totalise le produit des colonnes Cost (Coût) et Quantity (Quantité) :

     ```
     SUM(OrderDetails.prod_cost*OrderDetails.order_qty)
     ```

Pour plus d'informations sur les expressions C-SQL, voir Chapitre 14, «Formules», à la page 125.

8. Définissez une ou plusieurs colonnes de dimension.
   - Cliquez sur **Ajouter une dimension** pour définir une colonne.
   - Choisissez la dimension à inclure dans la liste déroulante de la colonne **Dimension**. Cette liste répertorie toutes les dimensions pour lesquelles vous disposez d'un accès en lecture seule au moins.
   - Identifiez les colonnes clés dans la dimension et dans la table de faits. Pour plus de détails sur les clés, voir «Colonnes clés», à la page 93.

   Le type de données associé à la clé dans la table de faits doit être identique à celle de la dimension (table de correspondance). Vous devez utiliser soit le type Décimal, soit le type Entier, mais pas un mélange des deux.
9. Enregistrez le cube.

Chapitre 9. Types de données

IBM Cognos Real-time Monitoring et C-SQL prennent en charge divers types de données SQL-99.

Tableau 28. Types de données C-SQL

Type de données C-SQL	Catégorie de type de données
BOOLEAN	«Opérateurs booléens», à la page 82
DECIMAL	«Types de données C-SQL pour les valeurs numériques», à la page 71
DOUBLE PRECISION	«Types de données C-SQL pour les valeurs numériques», à la page 71
INTEGER	«Types de données C-SQL pour les valeurs numériques», à la page 71
INTERVAL	«Date-heure», à la page 76
LONG	«Types de données C-SQL pour les valeurs numériques», à la page 71
TIMESTAMP	«Date-heure», à la page 76
VARCHAR	«Chaîne», à la page 74

C-SQL offre différentes méthodes pour convertir les données d'un type en un autre type. Pour en savoir davantage, voir «Conversion des types de données».

Conversion des types de données

IBM Cognos Real-time Monitoring vous offre deux moyens de convertir des valeurs d'un type de données vers un autre : conversion explicite et conversion implicite.

Conversion explicite

N'importe quel argument C-SQL peut contenir la commande CAST() pour convertir le type de données d'une valeur.

Pour plus d'informations, voir «CAST», à la page 142.

Par exemple, vous pouvez convertir une chaîne de chiffres en une valeur numérique et utiliser le résultat en tant qu'argument de FLOOR (voir «FLOOR», à la page 155) :

```
FLOOR( CAST( '1234.56' AS DECIMAL) )
```

Conversion implicite

C-SQL tente automatiquement de convertir un type de données en un type adéquat pour l'argument dans lequel la valeur est utilisée.

Par exemple, si C-SQL rencontre la valeur VARCHAR '1234.56' dans l'argument FLOOR(), il la convertit automatiquement en une valeur numérique DOUBLE PRECISION avant de tronquer les décimales. Par exemple :

© Copyright IBM Corp. 2007, 2014

```
FLOOR( '1234.56' )    << Implicit cast to DOUBLE
PRECISION.
```

De la même façon, lorsqu'une valeur d'un type de données est comparée à une valeur d'un autre type, C-SQL commence par convertir l'une des valeurs pour lui attribuer le même type de données que l'autre. Dans l'exemple ci-après, C-SQL convertit la chaîne VARCHAR en une valeur BOOLEAN avant d'évaluer l'expression :

```
'true' = TRUE         << Implicit cast to BOOLEAN.
```

Le contexte se répercute également sur la conversion. Par exemple, étant donné que l'opérateur arithmétique d'addition ci-après attend des arguments numériques, et bien que les deux valeurs soient des caractères, ces dernières sont d'abord converties en valeurs numériques :

```
'2' + '3'         << Both cast to numeric to match
operator data type.
```

Si la valeur n'est pas convertible dans le type requis, l'expression produit un message d'erreur type de données incorrect.

Ordre de priorité

Chaque conversion de type de données est associée à un ordre de priorité.

Le tableau ci-après présente les types de données convertibles, ainsi que l'ordre de priorité affecté à chacune des conversions possibles, où zéro (0) indique la priorité la plus élevée et un million (1 000 000) la priorité la plus faible :

Tableau 29. Ordre de priorité affecté à chaque conversion de type de données

Vers/Depuis	VARCHAR	BOOLEAN	TIMESTAMP	DOUBLE PRECISION	DECIMAL	INTEGER	LONG
VARCHAR	0	1 000 000	1 000 000	1 000 000	1 000 000	1 000 000	10
BOOLEAN	1	0					
TIMESTAMP	2		0				
DOUBLE PRECISION	3			0	1	3	2
DECIMAL	4			2	2	2	1
INTEGER	5			4	0	0	3
LONG	6			3	1	1	0

Dans l'exemple de comparaison fourni à la section «Conversion implicite», à la page 69, C-SQL convertit la valeur VARCHAR en une valeur BOOLEAN car le niveau de priorité de cette conversion est de 1, contrairement à la conversion du type BOOLEAN en type VARCHAR dont le niveau de priorité est de 10. De la même façon, dans l'exemple FLOOR("1234.56"), C-SQL convertit la chaîne en valeur DOUBLE PRECISION, car le niveau de priorité de ce type de données est plus élevé que celui du type DECIMAL, bien que ce dernier type semble plus adéquat pour cette valeur.

Pour plus de détails sur la conversion des types de données C-SQL, reportez-vous aux descriptions des différents types.

Types de données C-SQL pour les valeurs numériques

C-SQL comporte plusieurs types de données pour les valeurs numériques.

Tableau 30. Types de données pour les valeurs numériques

Type	Description
INTEGER	Entier signé 32 bits -2,147,483,648 (valeur minimale) 2,147,483,647 (valeur maximale)
DECIMAL	Décimal 1 chiffre minimum 256 chiffres maximum Le nombre total de chiffres DECIMAL, avant et après le séparateur décimal, est 256.
DOUBLE PRECISION	Virgule flottante IEEE 754 64 bits +/-4.94065645841246544E-324 (valeur minimale) +/-1.79769313486231570E-308 (valeur maximale) Pour exprimer une valeur DOUBLE PRECISION en tant que littéral, utilisez une notation scientifique, par exemple, 1e24.
LONG	Long signé 64 bits -2^{64} (valeur minimale) 2^{64} - 1 (valeur maximale)

Avertissement :

- Le nombre total de chiffres DECIMAL, avant et après le séparateur décimal, est 256.
- Pour exprimer une valeur DOUBLE PRECISION sous la forme d'un littéral, utilisez une notation scientifique, telle que 1e24.

Types de données tiers correspondant aux types de données numériques C-SQL

Les types de données numériques C-SQL correspondent aux types de données d'autres systèmes pris en charge.

Le tableau ci-dessous explique le mappage des données numériques C-SQL en types de données d'autres systèmes pris en charge, dans lequel *p* signifie précision :

Tableau 31. Mappage des données numériques C-SQL aux types de données d'autres systèmes

Système pris en charge	Entier	Long	Decimal	Double
C-SQL/ JDBC	INTEGER	LONG	DECIMAL	DOUBLE PRECISION
Java	int	long	BigDecimal	double
Oracle	Number(*p*=38)	Number((19,0)	Number	Number(*p*=38)
SQL Server	Int(32 bits)	BigInt	Decimal Numeric Money SmallMoney	double real(4 bytes)
Sybase	Int(32 bits)	BigInt	Decimal(*p*=38) Numeric(*p*=38) Money(64 bits) SmallMoney(32 bits)	double real(4 bytes)
MySQL	TINYINT SMALLINT MEDIUMINT INT, INTEGER BIGINT	BIGINT	DECIMAL NUMERIC	DOUBLE REAL
PostgreSQL	SMALLINT INT, INTEGER BITINT	BIGINT	DECIMAL NUMERIC	DOUBLE PRECISION REAL

Combinaison de types numériques

Lorsque vous combinez deux types numériques distincts, le résultat correspond au type qui possède la priorité la plus élevée. Par exemple, l'addition d'une valeur INTEGER et d'une valeur DECIMAL produit un total de type DECIMAL.

La priorité est établie selon l'«Ordre de priorité», à la page 70.

La conversion d'un nombre fractionnaire en nombre entier tronque la fraction (en arrondissant la valeur au chiffre inférieur) pour produire le résultat attendu. Par exemple, la conversion d'une valeur DOUBLE PRECISION en valeur INTEGER tronque la partie fractionnaire de la valeur.

Conversion de types numériques

La conversion de valeurs numériques en types dotés d'une taille de stockage distincte est autorisée à condition que la cible ait une capacité suffisante pour

contenir le résultat ; dans le cas contraire, la conversion échoue et renvoie un message d'erreur du type "valeur non comprise dans la plage autorisée". Par exemple, toute tentative d'insertion d'un type de valeur à virgule flottante dans un emplacement doté d'une taille de stockage plus restreinte génère une erreur.

Conversion de types numériques en chaînes

Lors de la conversion de valeurs numériques en chaînes, vous devez tenir compte de certaines pratiques.

Ces pratiques sont les suivantes :

- Dans le cas d'un nombre DECIMAL, des zéros sont ajoutés à cette valeur décimale de façon à correspondre à la précision et à l'échelle définies pour la colonne. Par exemple, si une colonne est définie avec une précision de 5 et une échelle de 4, la valeur 1.1 de la colonne devient ‘1.1000’.
- Pour les nombres à DOUBLE PRECISION, le 'e' est converti en majuscule. Par exemple, +1e11 est converti en ‘1.0E11’.

Précision décimale des résultats

Tous les nombres décimaux se composent de deux éléments : la précision et l'échelle.

- Précision

  Nombre de chiffres à gauche et à droite du signe décimal. Le nombre maximal est de 256 et le nombre minimal de 1.
- Echelle

  Nombre de chiffres de la partie fractionnaire. Ce nombre est inférieur ou égal à la précision. Lorsque aucune échelle n'est spécifiée, la valeur par défaut est de 2.

Dans les cas où une valeur est dotée d'une échelle ou d'une précision supérieure à celle de la valeur de stockage cible, telle qu'une zone de base de données dotée d'une précision plus réduite, IBM Cognos Real-time Monitoring tronque la valeur décimale et arrondit le résultat à la valeur inférieure pour l'adapter à la capacité de la cible.

Conversion

Lorsque vous convertissez une valeur décimale, vous pouvez déclarer la précision et l'échelle.

Vous définissez la précision et l'échelle de la manière suivante :

```
DECIMAL( precision, scale )
```

Par exemple :

```
CAST( '4.012345', DECIMAL(5,4) ) --> 4.0123
```

En cas de conversion des valeurs décimales d'une colonne sous forme de chaîne, des zéros sont ajoutés aux valeurs décimales de façon à correspondre à l'échelle. Par exemple, si une colonne est définie avec une précision de 5 et une échelle de 4, la conversion implicite de la valeur 1.1 de la colonne en une chaîne produit le résultat '1.1000'.

Multiplication

Dans une multiplication, la précision résultante est la somme des précisions, et l'échelle correspond à la somme des échelles.

Les résultats sont obtenus à partir du calcul suivant :

```
PrecisionResult = MIN( PrecisionLeft+PrecisionRight,256)
ScaleResult = MIN( ScaleLeft+ScaleRight, 256)
```

Par exemple, le résultat de (4.55*1.414) présente la précision **7** (3+4) et l'échelle **5** (2+3).

Division

Le résultat des divisions est obtenu à partir du calcul ci-dessous.

Par exemple, le résultat de (4.55/1.414) est l'échelle **4** (2+4-1) et la précision **10** (3+3+4). Le calcul est le suivant :

```
ScaleResult = MIN(MAX((ScaleLeft+PrecisionRight-ScaleRight+1),2),256)
PrecisionResult = MIN((PrecisionLeft + ScaleRight + ScaleResult),256)
```

Addition et soustraction

Le résultat des additions et des soustractions est obtenu à partir du calcul ci-dessous.

```
PrecisionResult = MIN( (MAX( PrecisionLeft - ScaleLeft,
                      PrecisionRight - ScaleRight) +
                  MAX( ScaleLeft, ScaleRight ) + 1),256)
ScaleResult = MAX( ScaleLeft, ScaleRight )
```

Autres fonctions

Pour toutes les autres fonctions et opérations, le résultat est déterminé par les valeurs dotées de la plus grande précision et de l'échelle maximale. Les résultats peuvent être obtenus à partir de la même valeur.

```
PrecisionResult = MIN( MAX( Precision[i] ), 256)
ScaleResult = MIN( MAX( Scale[i] ), 256)
```

Chaîne

Le type de données C-SQL VARCHAR conserve les valeurs de chaîne de caractères.

Tableau 32. Type de données C-SQL VARCHAR

Type	Description	Caractères de remplissage	Taille minimale	Taille maximale
VARCHAR	Longueur variable	Non	1 caractère (valeur par défaut). Peut avoir la valeur NULL.	Nombre de caractères illimité. Notez que vous obtenez une erreur si vous tentez de stocker une valeur dans un SGBD plus volumineux que la taille de la colonne définie dans la table.

Largeur des chaînes

Bien que la taille maximale des valeurs de chaîne soit illimitée, évitez de dépasser 255 caractères car cette limite est imposée par un grand nombre de systèmes de gestion de base de données.

Toutefois, pour optimiser les performances, simplifier le stockage des données et faciliter les comparaisons de chaînes, il est conseillé de déclarer une largeur maximale appropriée lors de la définition d'une colonne VARCHAR. Cette largeur doit être suffisamment importante pour prendre en charge la longueur maximale de n'importe quelle chaîne résultante insérée dans la zone. Tout texte plus long que la largeur maximale est tronqué lors du stockage de la chaîne.

Types de données tiers correspondant au type VARCHAR

Le type de chaîne VARCHAR C-SQL/JDBC est mappé à des types de données d'autres systèmes pris en charge.

Le tableau ci-dessous explique le mappage du type de chaîne VARCHAR C-SQL/JDBC en types de données d'autres systèmes pris en charge :

Tableau 33. Mappage du type de chaîne C-SQL/JDBC VARCHAR aux types de données d'autres systèmes

Java	Oracle	SQL Server	Sybase	MySQL	PostgreSQL
Chaîne	Char Varchar Varchar2(4k)	Varchar(8k)	Varchar(8k)	CHAR VARCHAR BINARY VARBINARY BLOB ENUM SET	CHAR[n] VARCHAR[n

Concaténation de chaîne

Vous pouvez concaténer deux chaînes.

Pour ce faire, utilisez la fonction CONCAT ou l'opérateur `||`. Pour en savoir davantage, voir «CONCAT», à la page 168.

Littéral de chaîne

Pour exprimer une chaîne sous la forme d'un littéral, indiquez le texte entre apostrophes (''). Pour insérer un guillemet simple, entrez deux apostrophes.

Par exemple :

```
'Couldn''t'     Returns: Couldn't
```

Conversion de chaînes dans d'autres types de données

Lorsque vous combinez une chaîne avec un type de données différent ou que vous exprimez une chaîne à un emplacement destiné à un autre type de données, la chaîne est automatiquement convertie dans le nouveau type en fonction de l'ordre de priorité.

Pour plus d'informations, voir «Ordre de priorité», à la page 70.

Lors de la conversion, les événements suivants se produisent :

- Tous les espaces de début et de fin sont supprimés.
- Si la chaîne contient un caractère ou une mise en forme non valide, une erreur survient. Un caractère non valide est un caractère inapproprié pour le type de données cible. Par exemple, vous ne pouvez pas convertir la chaîne 'bonjour' en une valeur de type INTEGER.
- Une mise en forme non valide est une mise en forme non cohérente avec la définition d'une valeur de donnée littérale du type cible. Par exemple, pour qu'une chaîne puisse être implicitement convertie en un type de données TIMESTAMP, la chaîne source doit être contenue dans le format de date C-SQL par défaut. Pour en savoir plus sur le formatage des chaînes de type date-heure, voir «Date-heure».

Date-heure

Les types de données date-heure stockent des données de date et d'heure sous la forme d'une valeur unique (nombre).

Il n'existe aucune fonctionnalité permettant d'accéder directement à une date-heure sous la forme de sa représentation numérique interne. Pour offrir un accès pertinent aux valeurs date-heure, C-SQL propose en revanche différentes fonctions de manipulation des valeurs, ainsi que des structures littérales permettant de représenter les valeurs dans des expressions.

Tableau 34. Types de données date-heure

Fonction	Description
Littéral TIMESTAMP (voir «Littéral TIMESTAMP», à la page 78)	Représentation d'une valeur date-heure sous la forme d'une chaîne de caractères. Il peut s'agir de n'importe quelle combinaison d'année, de mois, de jour, d'heures, de minutes, de secondes et de secondes fractionnaires.
Littéral INTERVAL (voir «Littéral INTERVAL», à la page 78).	Représentation d'un intervalle sous la forme d'une chaîne de caractères : plage de temps composée d'années et de mois ou de jours, d'heures, de minutes et de secondes.

Les valeurs date-heure appartiennent au fuseau horaire des paramètres régionaux du serveur exécutant IBM Cognos Real-time Monitoring.

Conversion des valeurs date-heure en chaînes

Il vous est possible de convertir une valeur date-heure en une chaîne de caractères (VARCHAR) ou une chaîne en valeur de données date-heure. Dans un cas comme dans l'autre, vous pouvez définir le format de la chaîne.

Convertissez une valeur date-heure en une chaîne de caractères (VARCHAR) par le biais de TO_CHAR (voir «TO_CHAR», à la page 150), puis convertissez une chaîne en une valeur date-heure par le biais de TO_DATE (voir «TO_DATE», à la page 151).

En insérant un littéral TIMESTAMP (voir «Littéral TIMESTAMP», à la page 78) dans une chaîne, un argument convertit automatiquement la valeur en une chaîne en utilisant le format de date-heure par défaut, à savoir yyyy-MM-ddhh:mm:ss.SSS. Pour plus d'informations sur la conversion des valeurs de chaîne et de date-heure, voir «Conversion des types de données», à la page 69.

Comparaison des valeurs date-heure

Une valeur date-heure est stockée de façon interne sous la forme d'un nombre représentant la date et l'heure en millisecondes. Vous devez donc faire preuve de prudence lorsque vous comparez deux valeurs date-heure.

Par exemple, la comparaison ci-après n'est vraie que si les deux dates comportent le même nombre de millisecondes :

```
first_date = second_date
```

Si une granularité exacte n'a pas d'importance, envisagez de commencer par convertir les valeurs date-heure en chaînes représentant uniquement la partie date :

```
TO_CHAR(first_date,"yyyy-MM-dd") = TO_CHAR(second_date,"yyyy-MM-dd")
```

En fonction de l'«Ordre de priorité», à la page 70, la comparaison d'une chaîne avec une valeur date-heure commence par convertir la chaîne en date-heure avant le début de la comparaison. Examinez l'exemple ci-après dans lequel birth_date est une valeur date-heure. Si la valeur birth_date est associée à une heure, la comparaison n'est jamais vraie :

```
'2010-02-18' = birth_date
```

Une comparaison plus exacte consiste à convertir la valeur birth_date en une chaîne n'indiquant aucune heure :

```
'2010-02-18' = TO_CHAR( birth_date, "yyyy-MM-dd" )
```

Arithmétique appliquée aux valeurs date-heure

Vous pouvez additionner et soustraire des intervalles d'années, de mois, de jours, d'heures, de minutes et de secondes sur des valeurs date-heure.

Pour de plus amples informations, reportez-vous à la description de ces fonctions. Toutefois, certaines clauses de requête nécessitent un littéral INTERVAL.

Pour plus d'informations sur le littéral INTERVAL, voir «Littéral INTERVAL», à la page 78.

Types de données tiers correspondant au type date-heure C-SQL

Le type date-heure C-SQL est mappé aux types de données d'autres systèmes pris en charge.

Le tableau suivant décrit les mappages du type date-heure.

Tableau 35. Mappages du type date-heure

Système pris en charge	Type date-heure
C-SQL/JDBC	Date-time
Java	Date Heure Horodatage
Oracle	Date(YMDHMS)
SQL-Server	Datetime(YMDHMS.xx) SmallDateTime (YMDHMS)
Sybase	Datetime (YMDHMS.xx) SmallDateTime (YMDHMS)
MySQL	DATE DATETIME TIMESTAMP TIME YEAR
PostgreSQL	TIMESTAMP

Littéral TIMESTAMP

Le littéral TIMESTAMP représente une valeur date-heure sous la forme d'une chaîne de caractères. Pour exprimer une valeur date-heure sous la forme d'une valeur littérale, faites précéder les données du préfixe "TIMESTAMP" et indiquez la totalité des données entre apostrophes (').

Par exemple :

```
TIMESTAMP '2010-03-05 19:45:23.123'
```

Le format de cette chaîne est "yyyy-MM-dd hh:mm:ss.SSS", où les valeurs S (les secondes fractionnaires) sont facultatives et dotées d'une précision comprise entre zéro et neuf chiffres. Pour plus d'informations sur les caractères de mise en forme, voir «Format date-heure», à la page 81.

Littéral INTERVAL

Un littéral INTERVAL identifie une plage de temps composée d'années et de mois (intervalles année-mois) ou de jours, d'heures, de minutes et de secondes (intervalles jour-heure).

Vous ne pouvez pas combiner un intervalle année-mois et un intervalle jour-heure dans une même déclaration d'intervalle. Les intervalles sont appliqués aux valeurs date-heure pour le calcul d'une plage de temps à partir de cette instance.

Ils sont généralement utilisés dans les expressions pour équilibrer les colonnes date-heure et les littéraux TIMESTAMP, notamment en cas de déclaration de la plage à partir d'une date ou d'une heure dans les fenêtres de requêtes. Pour plus d'informations, voir Chapitre 38, «Fenêtres de requêtes», à la page 391. Par exemple, la fenêtre de requête ci-après totalise tous les événements survenus au cours de la dernière heure (implicitement appliquée à l'heure d'arrivée du tout dernier événement) :

```
SUM(Qty) AS Total_Of_Qty OVER (RANGE INTERVAL
'1' HOUR PRECEDING)
```

Lors de l'application d'un intervalle à une valeur date-heure, l'intervalle est ajouté à cette valeur ou retranché de cette dernière. Par exemple, si la date-heure du jour correspond au 5 mars 2003 à 19:45, l'ajout d'un intervalle d'une année à cette date renvoie la valeur 5 mars 2004 à la même heure. Les opérations arithmétiques appliquées au calendrier suivent les règles du calendrier grégorien. Pour plus d'informations, voir «DATE_DIFF», à la page 145.

Intervalles année-mois

Une valeur INTERVAL année-mois utilise l'une des zones date-heure YEAR ou MONTH, ou combine les deux.

Les définitions possibles sont les suivantes :

```
INTERVAL 'yy'    YEAR[(<precision>)]
INTERVAL 'mm'    MONTH[(<precision>)]
INTERVAL 'yymm' YEAR[(<precision>)] TO
MONTH[(<precision>)]
```

Les exemples ci-après définissent des intervalles de trois ans et 10 mois :

```
INTERVAL '3'  YEAR
INTERVAL '10' MONTH
```

Vous pouvez définir un intervalle d'années fractionnaire en exprimant le résultat en nombre total de mois, tel que 46 mois, ou en combinant la zone. L'exemple suivant identifie un intervalle de 3 ans et 10 mois :

```
INTERVAL '3-10' YEAR TO MONTH
```

Vous pouvez définir une valeur zéro (0) pour l'une ou l'autre zone. Chacun des intervalles ci-après est de deux ans :

```
INTERVAL '2-0'  YEAR TO MONTH
INTERVAL '1-12' YEAR TO MONTH
INTERVAL '0-24' YEAR TO MONTH
```

Précision des intervalles année-mois

L'argument <précision> est un standard ANSI qui déclare le nombre maximal de chiffres du nombre entier. La <précision> par défaut est de 2.

Dans ce cas, les deux déclarations d'intervalles de 100 mois ci-après échoueront :

```
INTERVAL '100' MONTH(2)  <<
ERROR, precision is less than value size.
INTERVAL '100' MONTH     << ERROR,default
precision is 2.
```

Pour utiliser plus de deux chiffres, déclarez la précision de la façon suivante :

```
INTERVAL '100' MONTH(3)
```

Si vous utilisez les deux zones, n'appliquez la précision qu'à la zone YEAR ; la zone MONTH applique une précision par défaut de 2. Par exemple, la déclaration ci-après est erronée, car le mois excède la valeur par défaut.

```
INTERVAL '100-123' YEAR(3) TO MONTH   << ERROR,month
is 3 digits
```

Intervalles jour-heure

Une valeur INTERVAL jour-heure est constituée d'une combinaison de jours, d'heures, de minutes et de secondes.

Les définitions possibles sont les suivantes (.nn constituant la partie fractionnaire d'une seconde) :

```
INTERVAL 'dd'                DAY
INTERVAL 'ddhh'              DAY TO HOUR
INTERVAL 'ddhh:mm'           DAY TO MINUTE
INTERVAL 'ddhh:mm:ss[.nn ]'  DAY TO SECOND
INTERVAL 'hh'                HOUR
INTERVAL 'hh:mm'             HOUR TO MINUTE
INTERVAL 'hh:mm:ss[.nn ]'    HOUR TO SECOND
INTERVAL 'mm'                MINUTE
INTERVAL 'mm:ss[.nn ]'       MINUTE TO SECOND
INTERVAL 'ss[.nn ]'          SECOND
```

Voici quelques exemples d'intervalles jour-heure :

```
INTERVAL '27 23:59:59.999999999' DAY TO SECOND
INTERVAL '100 10:10' DAY(3) TO MINUTE
```

Précision des intervalles jour-heure

Chacune des zones de jour-heure comporte également un argument de précision.

Par exemple :

```
MINUTE(<precision>)
```

L'argument <précision> est un standard ANSI qui déclare le nombre maximal de chiffres du nombre entier. La <précision> par défaut est de 2 (sauf pour les secondes fractionnaires dont la précision par défaut est de 9 ; pour plus de détails, voir l'exemple ci-dessous). Dans ce cas, les deux déclarations d'intervalles de 100 heures ci-après échoueront :

```
INTERVAL '100' HOUR(2)  << ERROR,precision
is less than value size.
INTERVAL '100' HOUR     << ERROR,default
precision is 2.
```

Pour utiliser plus de deux chiffres, déclarez la précision de la façon suivante :

```
INTERVAL '100' HOUR(3)
```

Pour déclarer la précision de la valeur SECOND avec une partie fractionnaire, définissez deux valeurs de précision en les séparant par une virgule. Examinez les exemples ci-après :

```
INTERVAL '12.345' SECOND(2, 3)
INTERVAL '12.123456789' SECOND(2, 9)
INTERVAL '12.123456789' SECOND
```

Notez que les deux derniers exemples ci-dessus ont le même effet, car la précision par défaut est de (2, 9) pour la valeur SECOND.

Si vous utilisez plusieurs zones, n'exprimez la précision que sur la première zone. Les zones restantes utilisent leur valeur par défaut. Par exemple, la précision de l'exemple ci-après ne s'applique qu'aux minutes et ne se répercute pas sur les secondes fractionnaires :

```
INTERVAL '100:23.123456789' MINUTE(3) TO SECOND
```

Format date-heure

Pour certaines fonctions, il vous est possible de définir le format de la chaîne date-heure.

Les fonctions TO_CHAR (voir «TO_CHAR», à la page 150) et TO_DATE (voir «TO_DATE», à la page 151) ont toutes les deux des arguments qui définissent le format de la chaîne date-heure. La chaîne modèle du format de date, illustrée dans le tableau ci-dessous, est identique à celle utilisée par la classe Java SimpleDateFormat, qui utilise les lettres suivantes dans ses modèles :

Tableau 36. Chaîne modèle du format de date utilisée par la classe Java SimpleDateFormat

Lettre	Composant date-heure	Présentation	Exemples
G	Indicateur d'ère	Texte	AD
y	Année	Année	2010; 10
M	Mois dans l'année	Mois	Juillet; Jul; 10
w	Semaine dans l'année	Numérique	27
W	Semaine dans le mois	Numérique	2
D	Jour dans l'année	Numérique	189
j	Jour dans le mois	Numérique	10
F	Jour de la semaine dans le mois	Numérique	2
E	Jour dans la semaine	Texte	Mardi; mar
a	Symboles AM / PM	Texte	PM
H	Heure dans une journée (0-23)	Numérique	0
k	Heure dans une journée (1-24)	Numérique	24
K	Heure am/pm (de 0 à 11)	Numérique	0
h	Heure am/pm (de 1 à 12)	Numérique	12
m	Minute dans l'heure	Numérique	30
s	Seconde dans la minute	Numérique	55
S	Fraction de seconde (un S renvoie toujours un entier de 0 à 9 chiffres)	Numérique	978
'	Traiter les caractères spéciaux comme du texte	Délimiteur	

Tableau 36. Chaîne modèle du format de date utilisée par la classe Java SimpleDateFormat (suite)

Lettre	Composant date-heure	Présentation	Exemples
''	Guillemet simple	Littéral	'

Certaines lettres peuvent avoir plusieurs résultats, en fonction du nombre de lettres consécutives dans le format. Le résultat est la valeur qui correspond le mieux au modèle. Pour les nombres, si le modèle est plus grand que la valeur, le résultat est complété par des zéros à gauche (0). Examinez les exemples du tableau ci-dessous pour en savoir plus.

Les exemples du tableau ci-dessous, modifiés à partir de la documentation de classe Java SimpleDateFormat, illustrent la manière dont les modèles de date et d'heure sont interprétés dans les paramètres régionaux US. La date et l'heure données sont 2010-08-04 12:08:56 heure locale aux Etats-Unis (Pacifique).

Tableau 37. Modèles de date et d'heure interprétés dans l'environnement local US

Modèle date-heure	Résultat
'd M yy'	4 8 10
'dd MM yy'	04 08 10
'ddd MMM yyy'	004 Aug 2010
'dddd MMMM yyyy'	0004 August 2010
"yyyy.MM.dd G 'at' HH:mm:ss z"	2010.08.04 AD à 12:08:56 PDT
"EEE, MMM d, ''yy"	Sat, Aug 4, '10
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02010.August.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Sat, 4 Aug 2010 12:08:56 -0700
"yyMMddHHmmssZ"	100704120856-0700

Opérateurs booléens

Le langage C-SQL suit l'utilisation SQL-99 de la logique à trois valeurs (TRUE, FALSE et UNKNOWN) pour prendre en charge la sémantique des valeurs NULL.

Par exemple :

```
WHERE OnSale IS TRUE
WHERE (Age >= 21) IS UNKNOWN
```

Table de vérité pour NOT

Lorsque vous utilisez des opérateurs booléens pour évaluer la vérité d'une expression, les valeurs sont évaluées conformément aux tables de vérité ci-après.

Tableau 38. Table de vérité pour NOT

NOT	TRUE	FALSE	UNKNOWN
	FALSE	TRUE	UNKNOWN

Table de vérité pour AND

Le tableau suivant est une table de vérité pour AND.

Tableau 39. Table de vérité pour AND

AND	**TRUE**	**FALSE**	**UNKNOWN**
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

Table de vérité pour OR

Le tableau suivant est une table de vérité pour OR.

Tableau 40. Table de vérité pour OR

OR	**TRUE**	**FALSE**	**UNKNOWN**
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

Table de vérité pour IS

Le tableau suivant est une table de vérité pour IS.

Tableau 41. Table de vérité pour IS

IS	**TRUE**	**FALSE**	**UNKNOWN**
TRUE	TRUE	FALSE	FALSE
FALSE	FALSE	TRUE	FALSE
UNKNOWN	FALSE	FALSE	TRUE

La valeur TRUE est supérieure à la valeur FALSE dans les comparaisons.

Chapitre 10. Flux de données

Les tables de flux de données reçoivent des événements. Il s'agit de données produites par des applications métier externes qui enregistrent les transactions, identifient les modifications relatives à l'état de l'activité et regroupent les détails des activités métier.

IBM Cognos Real-time Monitoring reçoit des événements dans les tables de flux de données. Ces événements guident le traitement interne de Cognos Real-time Monitoring.

Les vues reposant sur les tables de flux de données agrègent ensuite les informations d'événement et génèrent les règles qui recherchent des conditions métier exceptionnelles.

Pour en savoir davantage, reportez-vous aux sections Chapitre 36, «Vues», à la page 363 et Chapitre 28, «Règles», à la page 321.

Cette section décrit le fonctionnement des tables de flux de données, les sources externes qu'elles prennent en charge, leurs propriétés ainsi que la façon de créer et de modifier les tables de flux de données.

Fonctionnement des flux de données

Les événements proviennent d'applications métier, de bases de données et de fichiers texte.

En général, les agents reçoivent ou récupèrent automatiquement les données d'événement et les chargent dans une table de flux de données. Vous pouvez également choisir de charger manuellement des données à partir de fichiers texte à l'aide d'IBM Cognos Real-time Monitoring Workbench. Les événements sont traités à leur arrivée et leurs données sont transmises aux vues métier. Les vues agrègent les données et peuvent alors récupérer des données de correspondance relatives à l'événement.

Figure 9. Fonctionnement des flux de données

© Copyright IBM Corp. 2007, 2014

Les événements sont traités dans l'ordre de leur réception dans le système. Lorsqu'un agent reçoit un grand nombre d'événements, tous les nouveaux événements reçus par d'autres agents sont mis en file d'attente derrière ce premier groupe et ne sont traités qu'une fois le traitement du premier groupe complètement achevé.

Sources d'événements externes pour les flux de données

Les agents peuvent accéder à plusieurs sources d'événements externes pour les flux de données.

Pour plus d'informations sur la manière dont IBM Cognos Real-time Monitoring fournit les agents permettant d'extraire et de recevoir des données d'événement, voir Chapitre 4, «Agents», à la page 25.

Certains événements sont transmis (envoyés) dans les tables de flux de données lorsqu'ils se produisent. D'autres sont chargés (extraits) dans les tables de flux de données suite à une demande, par exemple d'une base de données ou d'un fichier texte. Le tableau suivant récapitule les sources d'événements disponibles et identifie l'agent source pris en charge ainsi que les données à transmettre à la table de flux de données ou à en extraire.

Tableau 42. Sources d'événements

Source externe	Agent	Envoi ou extraction de flux de données
Infrastructure d'événements communs	Aucun. (Voir Chapitre 7, «Infrastructure d'événements communs», à la page 57)	Envoi
Java Messaging Service (JMS)	Java Messaging Service (JMS) (voir Chapitre 17, «Java Messaging Service (JMS)», à la page 221)	Envoi
Fichier texte (XML ou à plat)	Fichiers à plat (voir Chapitre 13, «Fichiers à plat», à la page 103)	Extraction
TIBCO Rendezvous (RV)	TIBCO Rendezvous (voir Chapitre 33, «TIBCO Rendezvous», à la page 349)	Envoi
Action HTTP Post	Envoi HTTP (voir Chapitre 16, «HTTP Post», à la page 215)	Envoi
Salesforce	Salesforce (voir Chapitre 29, «Salesforce», à la page 327)	Extraction
Service Web	Aucun. (Voir «Flux de données de service Web», à la page 377.)	Envoi

Attributs de flux de données

Chaque table de flux de données comporte un attribut de nom, de description et d'état, ainsi qu'un agent la plupart du temps.

Le tableau ci-dessous décrit les attributs :

Tableau 43. Attributs de flux de données

Attribut	Description
Nom	Identifie la table. Il s'agit du nom utilisé par les vues métier qui dépendent de cette table. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Enregistrer dans	Indique le dossier dans lequel la table de flux de données sera stockée. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Description	Description facultative qui peut contenir tous les caractères de texte.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Agent	Agent chargé de recevoir ou de récupérer les informations de flux de données et de transmettre ces dernières à la table de flux de données correspondante. Pour plus d'informations sur les types d'agent, voir Chapitre 4, «Agents», à la page 25.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Si les événements peuvent être traités dans n'importe quel ordre, désactivez cet attribut. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.

Création de tables de flux de données

Vous pouvez créer une table de flux de données afin de recevoir des événements.

Conseil :

Utilisez l'option **Traiter toutes les lignes de l'ensemble de résultats comme un seul événement** uniquement lorsque aucune autre option n'est disponible. Bien que l'activation de cette option augmente le nombre de lignes traitées par seconde,

elle peut avoir des répercussions négatives sur la réponse utilisateur, notamment sur le temps d'actualisation de l'interface utilisateur. De plus, lorsque vous activez cette option, un plus grand volume de mémoire transitoire est nécessaire, de sorte que si la limite de mémoire physique du système est atteinte, le nombre de lignes traitées est également réduit.

L'option **Toujours supprimer l'état pour les nouvelles données** a pour but de supprimer l'état à un moment spécifique de la journée ou du mois. Cependant, lorsque vous l'utilisez en même temps que l'option **Traiter toutes les lignes comme un seul événement** (qui ne s'appuie pas sur les capacités de flot de données de Cognos Real-time Monitoring), un grand nombre de lignes sont lues et traitées plusieurs fois. L'utilisation simultanée de ces deux options a une incidence sur les performances et doit être évitée.

Lorsque vous spécifiez un type de source pour un flux de données, notez que chacun des types comporte des attributs qui lui sont propres. Pour plus d'informations, consultez les sections suivantes :

- Chapitre 13, «Fichiers à plat», à la page 103
- Chapitre 16, «HTTP Post», à la page 215
- Chapitre 17, «Java Messaging Service (JMS)», à la page 221
- Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227
- Chapitre 29, «Salesforce», à la page 327
- Chapitre 33, «TIBCO Rendezvous», à la page 349
- Chapitre 37, «Services Web», à la page 377

Avant de commencer

Pour créer une table de flux de données, vous devez disposer d'une autorisation de création sur les tables et d'une autorisation en lecture seule sur l'agent qui va alimenter la table.

Pour plus d'informations sur l'autorisation de création, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer nouveau**.
4. Sélectionnez **Flux de données**.
5. Choisissez le type de source.
6. Renseignez les zones du formulaire de nouveaux flux de données.
7. Enregistrez l'objet avec l'état activé.

   Il devient ainsi prêt à recevoir des données.

Modification des tables de flux de données

Vous pouvez modifier les attributs des tables de flux de données.

Avertissement : La modification des attributs d'une table de flux de données entraîne la perte de l'état de l'objet et invalide éventuellement les vues dépendantes. Par exemple, si vous supprimez une colonne, toutes les vues ou règles faisant référence à cette colonne deviennent incorrectes. Toutefois, si vous redéfinissez la colonne dans la table, les vues dépendantes redeviennent automatiquement valides.

Notez que chacun des types de flux de données comporte des attributs qui lui sont propres. Pour plus de détails, consultez les sections suivantes :

- Chapitre 13, «Fichiers à plat», à la page 103
- Chapitre 16, «HTTP Post», à la page 215
- Chapitre 17, «Java Messaging Service (JMS)», à la page 221
- Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227
- Chapitre 29, «Salesforce», à la page 327
- Chapitre 33, «TIBCO Rendezvous», à la page 349
- Chapitre 37, «Services Web», à la page 377

Avant de commencer

Pour modifier une table de flux de données, vous devez disposer d'une autorisation en lecture et en écriture pour les tables et d'un accès en lecture seule à l'agent qui alimente la table. Pour en savoir davantage, voir «Droits d'accès», à la page 295.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez **Flux de données**.
3. Sous **Nom de l'objet**, sélectionnez le flux de données à modifier.
4. Dans le menu **Activités**, cliquez sur **Modifier**.
5. Modifiez les définitions, si nécessaire.
6. Enregistrez l'objet avec l'état activé.

   Il devient ainsi prêt à recevoir des données.

Chapitre 11. Dimensions

Les dimensions sont des ordres de classement de données associées et sont utilisées par les cubes pour classer des mesures.

Une dimension est un ordre de classement qui, pour chacun des niveaux, du plus élevé au plus bas, décrit des ensembles de données associées de plus en plus petits. Le tableau ci-après présente quelques exemples de dimensions dans lesquels le niveau supérieur de chaque liste contient l'ensemble d'éléments associés le plus vaste, et le niveau inférieur contient l'ensemble d'éléments le plus réduit.

Tableau 44. Exemples de dimensions

Heure	Géographie	Inventaire	Sécurité	Taxinomie
année	continent	classification	type	règne
trimestre	pays	type	classement	embranchement
mois	région	fabricant	société	classe
semaine	département	modèle	CUSIP	ordre
jour	comté	configuration		famille
heure	ville			genre
minute	quartier			espèce

Les cubes classent les mesures par dimension, puis par niveau et par valeur dans chaque dimension. Par exemple, une dimension d'emplacement peut filtrer les résultats pour afficher les mesures relatives à toutes les villes d'un état. L'illustration ci-après présente plusieurs emplacements filtrés par nom d'état, ce qui restreint les mesures à trois villes seulement.

Figure 10. Emplacements filtrés par nom d'état

© Copyright IBM Corp. 2007, 2014

Les dimensions tirent leurs valeurs des tables de correspondance : chaque niveau de dimension constitue une colonne de la table, et chaque ligne correspond à un élément de dimension unique. Dans l'exemple illustré ci-dessus, les noms de région apparaissent dans une colonne, les états dans une autre et les villes dans une troisième. Lorsque vous créez une dimension, vous identifiez la table de correspondance source existante ainsi que les colonnes à inclure, puis vous organisez les colonnes dans la hiérarchie de niveaux.

Hiérarchie de niveaux

La hiérarchie de niveaux permet d'accéder à un niveau de dimension supérieur ou inférieur dans les cubes. Lorsqu'un utilisateur visualise les données d'un niveau, il peut choisir de consulter le niveau de mesures immédiatement supérieur ou inférieur.

Par exemple, si l'utilisateur examine les résultats relatifs à un état, il peut choisir d'accéder au niveau supérieur pour afficher la mesure portant sur tous les états de la région. Il a également la possibilité de cliquer sur les résultats pour obtenir les résultats associés à chacune des villes de l'état. La hiérarchie de niveaux définit les niveaux de la dimension.

L'illustration ci-après présente quatre colonnes du fichier source, mais seulement trois de ces colonnes contiennent des données de niveau de dimension. Ces trois colonnes sont sélectionnées et organisées dans la hiérarchie, depuis le niveau le plus général jusqu'au niveau le plus spécifique.

Si vous sélectionnez **Utiliser cette dimension pour les graphiques géographiques**, une colonne Catégories géographiques s'affiche. Pour plus d'informations sur ces catégories, voir «Catégories géographiques», à la page 93.

Noms d'alias dans les cubes

Après avoir identifié les colonnes, vous pouvez éventuellement affecter des noms d'alias que les utilisateurs verront lorsqu'ils travailleront avec le cube.

Zone Trier par colonne

Dans IBM Cognos Real-time Monitoring Dashboard, les valeurs de dimension sont présentées dans l'ordre dans lequel elles ont été triées (défini par le serveur).

Par exemple, une liste de noms de mois apparaît dans l'ordre alphabétique, c'est-à-dire en commençant par April et en finissant par September, et non dans l'ordre chronologique. Pour définir un autre ordre, utilisez la zone Trier par colonne. Cette zone identifie une autre colonne contenant les valeurs à utiliser pour le tri. Par exemple, au lieu d'utiliser la colonne correspondant au nom de mois, utilisez la colonne du numéro de mois.

En outre, si la zone Trier par colonne vous présente plusieurs valeurs correspondant à une seule valeur de la dimension, l'ordre de tri est configuré sur la valeur minimale de la zone Trier par colonne. Si tel est le cas, sélectionnez avec soin la colonne à utiliser pour trier les valeurs. Par exemple, l'utilisation des noms de mois comme ordre de tri pour les trimestres risquerait de produire des résultats imprévisibles, car les mois seraient triés par ordre alphabétique et non par ordre numérique.

Catégories géographiques

Les catégories géographiques permettent d'utiliser des informations géographiques dans les cubes.

Le paramétrage d'une catégorie géographique spécifie le niveau de précision utilisé par IBM Cognos Real-time Monitoring Dashboard pour la colonne. Par exemple, si une colonne contient des noms d'état ou de province, vous devez lui affecter la catégorie géographique Etat ou province.

Les catégories disponibles sont les suivantes :

- Non utilisé
- Pays ou région
- Etat ou province
- Ville
- Rue
- Code postal

Utilisez la catégorie **Non utilisé** pour une colonne que vous ne souhaitez pas représenter sur un graphique géographique ou qui ne correspond à aucune des catégories disponibles. Par exemple, une colonne REGION_NAME contenant des données telles que North, South, East et West ne peut pas être associée à l'un des emplacements géographiques proposés ; vous devriez donc lui affecter la catégorie géographique **Non utilisé**.

Colonnes clés

Pour utiliser la table de correspondance en tant que dimension, vos données d'événement doivent identifier l'élément de dimension unique (ligne) auquel elles appartiennent.

Dans les données de table de correspondance d'emplacement, la valeur la plus unique de chaque élément correspond au nom de ville ; toutefois, bien que ce nom soit utilisable à des fins d'identification, il est préférable d'utiliser un numéro pour des raisons de performances.

En conséquence, les données de correspondance des emplacements doivent être identifiées par un nombre entier, comme ci-après :

```
region_id  region_name region_state  region_city
---------  ----------- ------------  ------------
        1  West        Nevada        Reno
        2  South       Texas         Austin
        3  East        New York      Rochester
        4  Central     Ohio          Toledo
        5  West        California    Pasadena
       ...
```

La table de faits qui fournit la valeur à mesurer pour le cube inclut alors également cette valeur clé pour identifier la dimension associée. Par exemple, l'enregistrement de commande ci-après est associé à Ojai, California, dans la région West :

```
order_id region_id total_sales...
-------- --------- -----------
```

```
102341        7      120000...
```

Le type de données correspondant à la clé de la table de faits doit être identique à celui de la clé de la dimension (table de correspondance). Vous ne pouvez pas, par exemple, mélanger les nombres entiers avec les nombres décimaux ; il faut utiliser l'un ou l'autre.

Lorsque vous définissez la dimension, identifiez une ou plusieurs zones clés utilisables pour déterminer le niveau spécifique. Vous pouvez choisir une colonne ne correspondant pas déjà à un niveau de dimension, car les niveaux sont automatiquement supposés constituer des clés potentielles. L'illustration ci-après ne présente qu'une seule colonne, REGION_ID, car toutes les autres colonnes de la dimension sont affectées à des niveaux. Lorsque vous définissez le cube par la suite, identifiez la colonne clé de la dimension correspondant à la clé de la table de faits.

Pour plus d'informations sur la définition des cubes, voir «Création de cubes», à la page 65.

Création de dimensions

Vous pouvez créer des dimensions à utiliser dans les cubes.

Conseil : Pour des performances optimales, activez la mise en cache pour la table de correspondance. Lorsque la mise en cache est désactivée, les performances des cubes sont susceptibles d'être fortement ralenties. Pour en savoir davantage, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

Avant de commencer

Avant de créer une dimension, vous devez disposer d'une autorisation de création sur les vues, les cubes et les dimensions et disposer d'au moins un accès en lecture seule sur la table de correspondance qui fournit les éléments de dimension.

Pour en savoir davantage, voir Chapitre 20, «Tables de correspondance», à la page 265.

Après avoir créé une dimension, vous pouvez immédiatement l'utiliser dans les cubes. Pour en savoir davantage, voir Chapitre 8, «Cubes», à la page 63.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Dimension**.
4. Attribuez un nom à la dimension dans la zone **Nom** et entrez éventuellement une description de la dimension.
5. Cliquez sur **Parcourir** pour accéder à la table de correspondance qui contient les éléments de dimension.
6. Dans la fenêtre Sélectionner un objet, choisissez l'objet à utiliser avec la **Table Fait**.
7. Cliquez sur **OK**.

8. Sélectionnez l'option **Utiliser cette dimension pour les graphiques géographiques** si vous prévoyez d'implémenter des graphiques géographiques. Cette opération ajoute une colonne **Catégories géographiques** à la liste **Zones sélectionnées**.
9. Définissez les niveaux de la hiérarchie :
   - Ajoutez à la hiérarchie des zones sélectionnées dans la liste **Zones disponibles**.
   - Classez les niveaux de l'ensemble du plus grand (en haut) au plus petit (en bas).
   - Identifiez une ou plusieurs colonnes clés à inclure.
   - Affectez éventuellement des noms d'alias aux niveaux.
   - Si vous le souhaitez, affectez des catégories géographiques à chacun des niveaux si vous avez coché la case **Utiliser cette dimension pour les graphiques géographiques**. Les catégories disponibles sont les suivantes : **Non utilisé**, **Pays ou région**, **Etat ou province**, **Ville**, **Rue** et **Code postal**.
10. Enregistrez la dimension.

Chapitre 12. Système de fichiers Cognos Real-time Monitoring

Le système de fichiers utilisé pour charger des données d'un fichier à plat dans une table de correspondance.

L'utilisation des fichiers à plat pour les données des tables de correspondance permet également d'extraire des données d'un fichier et de les stocker dans une base de données locale via l'option Bibliothèque JDBC.

Pour plus d'informations, voir «Tables de correspondance établies à partir de fichiers à plat».

Tables de correspondance établies à partir de fichiers à plat

L'agent de système de fichiers charge les données à partir d'un fichier à plat dans une table de correspondance via le mode de traitement par lots.

En fonction du calendrier d'invalidation, l'agent de système de fichiers lit périodiquement le fichier à plat. Lorsque le fichier est lu, l'agent récupère les données et les transmet à la table de correspondance pour traitement.

Figure 11. Flux de travaux pour les fichiers à plat

Les tables de correspondance établies à partir de fichiers à plat reçoivent les données en les téléchargeant depuis un fichier à plat dans IBM Cognos Real-time Monitoring Workbench. Les deux formats de fichiers à plat pris en charge par les tables de correspondance sont les suivants :

- Fichiers délimités
- Fichiers à largeur fixe

Pour en savoir plus sur ces types de fichiers, voir Chapitre 13, «Fichiers à plat», à la page 103.

Avant de créer une table de correspondance à partir des données d'un fichier à plat, vous devez disposer d'une autorisation de création sur les tables et d'une autorisation en lecture seule sur l'agent qui va alimenter la table.

Pour les fichiers délimités et à largeur fixe, vous pouvez avoir besoin d'un fichier qui contient des données au format des données réelles pour la table de correspondance. Utilisez cet exemple lorsque vous créez la table de correspondance afin de vous assurer que les zones sont correctement mappées à la table. L'utilisation d'un exemple de fichier est facultative. Vous pouvez créer la table de

© Copyright IBM Corp. 2007, 2014

correspondance sans aucune source. Toutefois, la source a été conçue pour vous aider à créer une table de correspondance.

Le tableau suivant décrit les attributs de base nécessaires à la création d'une table de correspondance à partir d'un fichier à plat. Ces attributs sont similaires à ceux des tables de flux de données avec les fichiers à plat. Pour plus d'informations sur les fichiers à plat, voir Chapitre 13, «Fichiers à plat», à la page 103.

Tableau 45. Attributs des tables de correspondance établies à partir de fichiers à plat

Attribut	Description
Nom	Identifie la table de correspondance. Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Indique le dossier dans lequel la table de correspondance doit être stockée. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Description	Description facultative qui peut contenir tous les caractères de texte.
Agent de système de fichiers	Agent de système de fichiers existant qui récupère les données d'une table de correspondance et d'un flux de données à partir d'un fichier texte. Créez un agent avec Cognos Real-time Monitoring Workbench. Pour plus d'informations, voir «Agent de système de fichiers», à la page 34.
Chemin racine	Chemin racine indiqué lorsque vous créez l'agent de système de fichiers.
Chemin d'accès relatif	Chemin d'accès relatif au chemin racine du fichier texte à lire pour les données de la table de correspondance.
Commencer l'importation à la ligne	Pour les fichiers délimités et à largeur fixe, identifie la ligne de texte dans le fichier source qui contient les premières données à importer. La valeur par défaut est 1, à savoir la première ligne. Utilisez cette option si le texte contient des informations d'en-tête ou de présentation inutiles.
Désactiver la table de correspondance lorsque ce nombre d'erreurs consécutives	Nombre d'erreurs consécutives reçues du système avant que celui-ci ne désactive la table de correspondance. Une fois désactivée, une table de correspondance doit être réactivée manuellement. La valeur par défaut est 5.
Utiliser cette ligne pour les noms de colonne	Lorsque vous utilisez un exemple de fichier pour les fichiers délimités et à largeur fixe, cette option identifie une ligne dans l'exemple qui inclut les noms de colonne. Ces noms identifient chaque colonne dans l'onglet **Informations sur la zone**. Lorsque cette option n'est pas sélectionnée, les noms par défaut sont Field1, Field2, etc.

Tableau 45. Attributs des tables de correspondance établies à partir de fichiers à plat (suite)

Attribut	Description
Sauter les lignes qui utilisent l'expression régulière	Pour les fichiers délimités et à largeur fixe, cette option identifie les lignes à ignorer dans le fichier source avant d'importer les données du flux de données. Par exemple, si le fichier comporte une ligne qui contient ???, il se peut que la source commence effectivement à la troisième ligne du fichier. Dans ce cas, vous indiquez ??? ainsi que toutes les lignes comportant cette expression régulière à ignorer.
Autoriser les lignes courtes	Pour les fichiers délimités uniquement. Pour en savoir plus sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Délimiteur	Pour les fichiers délimités uniquement. Pour en savoir plus sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Caractère d'échappement	Pour les fichiers délimités uniquement. Pour en savoir plus sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Qualificatif de texte	Pour les fichiers délimités uniquement. Pour en savoir plus sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Formats numériques	Spécifications relatives au formatage des nombres. Les valeurs par défaut sont la virgule (,) comme séparateur de milliers et le point (.) comme séparateur décimal.

Onglet Informations sur la zone

Pour les fichiers délimités et à largeur fixe, utilisez cet onglet pour spécifier des informations sur chaque colonne de la table, y compris le nom, le type de données et le format applicable au type.

Tableau 46. Attributs des colonnes de table

Attributs	Description
Colonne	Cette zone permet de remplacer les noms de colonne par défaut (Field1, Field2, etc) ou de corriger le nom de la zone importée depuis un fichier d'exemple.
Index	Lorsque cette option est sélectionnée, un index est généré pour la zone. Il est essentiel de sélectionner l'index approprié afin de bénéficier de performances raisonnables lorsque la bibliothèque JDBC est activée pour la mise en cache des données. Sélectionnez **Index** pour les colonnes que vous utilisez dans les conditions de jointure.
Type de données	Indique le type de données pour la colonne. Le type de données de la colonne sélectionnée doit correspondre à celui de l'autre colonne utilisée pour la jointure.
Mise en forme	Indique le formatage du type de données. La mise en forme est applicable uniquement aux types de données DECIMAL, TIMESTAMP et VARCHAR. Pour modifier le format, sélectionnez **<Changer la mise en forme>** dans la liste déroulante.

Onglet Mise en cache des données

Cet onglet permet de définir les options de mise en cache.

Les paramètres de mise en cache sont les suivants : Type de cache, Extraire et stocker au redémarrage, Recourir à une source externe, Nombre de résultats à mettre en cache et Calendrier d'invalidation.

Type de cache

Vous avez le choix entre la mise en cache en mémoire ou la bibliothèque JDBC.

- **En mémoire**

  Lorsque vous activez la mise en cache en mémoire, la table de correspondance externe tout entière est mise en cache en mémoire à partir de la source de données externe et devient une solution de rechange pour la source de requête externe. L'activation de la mise en cache en mémoire revient au même que si vous sélectionnez la mise en cache prefetch pour les sources de la table de correspondance à partir des agents JDBC et de service Web, comme décrit dans «Mise en cache prefetch», à la page 270.

  Si vous sélectionnez cette option, l'option **Bibliothèque JDBC** n'est pas disponible et la zone **Agent JDBC** est désactivée.

- **Bibliothèque JDBC**

  Lorsque vous sélectionnez **Bibliothèque JDBC**, les données de la table de correspondance sont extraites vers une base de données locale, comme spécifié par l'**Agent JDBC**.

  La liste déroulante **Agent JDBC** présente les agents disponibles. Sélectionnez l'agent à utiliser pour la base de données locale.

Extraire et stocker au redémarrage

Cette option indique que lorsqu'IBM Cognos Real-time Monitoring est redémarré, le système tente toujours de récupérer les données de la source de requête externe, puis les stocke dans la base de données locale. Si une bibliothèque locale déjà alimentée existe, elle est utilisée jusqu'au prochain calendrier d'invalidation. Sinon, la source externe est interrogée au redémarrage.

Recourir à une source externe

Cette option indique que si le système ne parvient pas à accéder à la base de données locale, il utilise les données provenant de la base de données externe. Cette option n'est pas disponible lors de la configuration d'une table de correspondance basée sur un agent de système de fichiers.

Nombre de résultats à mettre en cache

Cette option détermine le nombre d'ensembles de résultats à mettre en cache. Chaque ensemble de résultats peut contenir une ou plusieurs lignes de données de table de correspondance liées au flux de données. Par exemple, si trois requêtes sont effectuées et que chaque ensemble de résultats comporte cinq lignes, 15 lignes sont stockées en cache.

Calendrier d'invalidation

Identifie le moment opportun pour invalider le cache et supprime toutes les informations contenues dans le cache, par exemple tous les lundis matin à 6 heures.

Suppression du contenu du cache

Vous pouvez ajouter un calendrier destiné à planifier la suppression du cache.

Procédure

1. Dans l'onglet **Mise en cache des données**, cliquez sur **Ajouter un calendrier**.
2. Programmez l'heure de suppression du cache.

Création d'une table de correspondance à partir d'un fichier à plat

Vous pouvez faire appel à un fichier à plat comme type source d'une table de correspondance.

Pourquoi et quand exécuter cette tâche

La procédure de création d'une table de correspondance à partir d'un fichier à plat est identique à celle qui permet de créer des sources de flux de données à partir de fichiers à plat. Pour plus de détails sur le type de source, voir :

- «Fichiers délimités», à la page 110.
- «Fichiers à largeur fixe», à la page 110.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**, puis sur le bouton **Activités**.
2. Sélectionnez **Créer**, puis **Table de correspondance**.
3. Sélectionnez **Système de fichiers** comme source de table de correspondance.
4. (Facultatif pour les fichiers délimités et à largeur fixe). Identifiez un fichier exemple qui va vous aider à mapper les colonnes. Ce fichier est un échantillon du fichier de données réel. Les données de ce fichier sont présentées à l'étape suivante afin de vous guider tout au long du processus de mappage des données de table de correspondance dans la table.
5. Pour les fichiers à largeur fixe, définissez l'emplacement des colonnes de données à l'aide de la boîte de dialogue Définir les largeurs des zones.
6. Définissez les informations sur la colonne en fonction du format choisi : fichiers délimités ou à largeur fixe.
7. Cliquez sur l'onglet **Mise en cache des données**.
8. Sélectionnez le **Type de cache**.
   - Lorsque vous activez le type de mise en cache **En mémoire**, cela revient au même que lorsque vous activez l'option **Activer prefetch** pour les sources de service Web et JDBC. Pour plus d'informations sur la mise en cache prefetch, voir «Mise en cache prefetch», à la page 270.
   - Lorsque vous activez le type de mise en cache **Bibliothèque JDBC**, cela signifie que la fonctionnalité d'extraction et de stockage est utilisée pour la mise en cache des données. Pour plus d'informations sur la fonctionnalité d'extraction et de stockage, voir «Mise en cache dans la bibliothèque JDBC (Java Database Connectivity)», à la page 271.
9. Enregistrez la source du fichier avec l'état activé.

Chapitre 13. Fichiers à plat

Un fichier à plat est un fichier texte contenant des informations sur un ou plusieurs événements ou sur les données d'une table de correspondance.

Pour les flux de données, chacune des lignes du fichier constitue généralement un enregistrement d'événement (une ligne de la table de flux de données), et les données de la ligne correspondent aux colonnes de la table de flux de données. Les lignes peuvent être présentées sous forme de fichiers à largeur fixe, délimités ou XML. Les fichiers délimités et à largeur fixe peuvent contenir plusieurs lignes pour les mêmes événements. Pour plus d'informations, voir «Evénements sur plusieurs lignes», à la page 107.

Pour les tables de correspondance, chacune des lignes du fichier constitue une ligne (ou un enregistrement). Les données de la ligne correspondent aux colonnes de la table de correspondance. Les lignes peuvent être présentées sous forme de fichiers à largeur fixe, délimités ou XML.

Données de fichiers à plat

Les données issues des fichiers à plat vous permettent de charger des données d'événement à partir d'un fichier texte dans une table de flux de données de fichier à plat. Cette section décrit le fonctionnement des données provenant des fichiers à plat ainsi que la procédure de création d'une table de flux de données à partir d'une source de fichier à plat.

Fonctionnement des fichiers à plat

Les événements d'un fichier à plat sont chargés en mode de traitement par lots dans la table de flux de données associée au fichier, mais sont traités individuellement lors de leur chargement dans la table.

Vous disposez de deux façons pour charger des fichiers à plat dans les tables de flux de données :

- Automatiquement

  L'agent de fichier à plat (voir «Agents de fichier à plat», à la page 30) vérifie régulièrement si le fichier associé existe. Lorsqu'un agent trouve le fichier, il le récupère et le transmet à l'objet de flux de données pour le traitement des flux de données.
- Manuellement

  Le bouton **Télécharger un fichier de données** situé sur la page de détails de chargement d'un fichier à plat dans la table de flux de données vous permet de charger un fichier sélectionné manuellement. Pour plus de détails, voir le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

© Copyright IBM Corp. 2007, 2014

Figure 12. Chargement des données dans une table de flux de données

Tables de flux de données associées aux fichiers à plat

Les tables de flux de données associées aux fichiers à plat reçoivent des fichiers de données à partir des agents de fichiers à plat ou en téléchargeant un fichier dans IBM Cognos Real-time Monitoring Workbench.

Les trois formats de fichiers à plat sont les suivants :

- «Fichiers délimités», à la page 110.
- «Fichiers à largeur fixe», à la page 110.
- «Prise en charge des fichiers XML», à la page 111.

Pour créer des données de table de flux de données à partir d'un fichier à plat, vous devez disposer au préalable des éléments suivants :

- Autorisations

  Autorisation de création (voir Chapitre 23, «Autorisations», à la page 295) sur les tables (voir «Autorisation de création d'objets», à la page 299) et autorisation Lecture seule sur l'agent qui alimente la table.
- Exemple de fichiers délimités et à largeur fixe (*facultatif*)

  Exemple de fichier qui contient des données au format des événements réels. Utilisez cet exemple lorsque vous créez l'objet de flux de données afin de vous assurer que les zones sont correctement mappées à la table de flux de données. Cet élément est facultatif. Vous pouvez créer l'objet de flux de données sans aucune source. Toutefois, l'exemple a été spécifiquement conçu pour vous aider à créer un objet de flux de données.
- Schéma des fichiers XML à charger

  Vous devez connaître le nom des attributs qui contiennent les informations liées à la colonne d'événement ainsi que le chemin d'accès XML à l'élément qui comporte les colonnes pour chaque événement. Pour plus de détails, voir «Emplacement XPath des colonnes et des lignes dans les fichiers XML», à la page 112. Le tableau ci-dessous décrit les différents attributs.

Tableau 47. Attributs d'information sur les colonnes

Attribut	Description
Nom	Identifie l'objet de flux de données. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Indique le dossier dans lequel l'objet de flux de données sera stocké. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Description	Description facultative qui peut contenir tous les caractères de texte.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*. Afin d'améliorer les performances d'entrée-sortie, choisissez un répertoire pour le fichier journal de reprise qui se trouve sur un disque différent de celui qui alimente l'agent.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Sinon, si les événements peuvent être traités dans n'importe quel ordre, désactivez cette option. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Agent de fichier	Agent de fichier existant qui récupère les données d'événement et de table de correspondance à partir d'un fichier texte. Créez un agent depuis l'onglet **Plan de travail**. Pour en savoir davantage, voir «Agents de fichier à plat», à la page 30.
Commencer l'importation à la ligne	Pour les fichiers délimités et à largeur fixe, identifie la ligne de texte dans le fichier source qui contient les premières données à importer. La valeur par défaut est 1, à savoir la première ligne. Utilisez cette option si le texte contient des informations d'en-tête ou de présentation inutiles.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive le flux de données lorsqu'un nombre d'erreurs consécutives se produisent. Par exemple, si la valeur définie est 5, le flux de données est désactivé au bout de 5 erreurs consécutives. Toutefois, dans le cas où 4 erreurs se produisent, puis 0 errer, puis 2 erreurs, le flux de données reste actif. La valeur par défaut est désactivée.

Tableau 47. Attributs d'information sur les colonnes (suite)

Attribut	Description
Utiliser cette ligne pour les noms de colonne	Lorsque vous utilisez un exemple de fichier pour les fichiers délimités et à largeur fixe, cette option identifie une ligne dans l'exemple qui inclut les noms de colonne. Ces noms identifient chaque colonne dans le détail des informations sur la colonne. Lorsque cette option n'est pas sélectionnée, les noms par défaut sont Field1, Field2, etc.
Sauter les lignes	Cette option identifie les lignes des fichiers délimités et à largeur fixe à ignorer dans le fichier source avant l'importation des données d'événement. Par exemple, si le fichier contient des informations de titre et d'en-tête, il se peut que la source comment effectivement à la troisième ligne du fichier. Dans ce cas, le nombre de lignes à ignorer sera 2.
Autoriser les lignes courtes	Pour les fichiers délimités uniquement. Pour plus de détails, voir «Fichiers délimités», à la page 110.
Traiter toutes les lignes de l'ensemble de résultats comme un seul événement	Toutes les lignes renvoyées dans l'ensemble de résultats sont considérées comme un événement. Sinon, chaque ligne renvoyée de la table est considérée comme un événement distinct.
Délimiteur	Pour les fichiers délimités uniquement. Pour plus d'informations sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Caractère d'échappement	Pour les fichiers délimités uniquement. Pour plus d'informations sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Qualificatif de texte	Pour les fichiers délimités uniquement. Pour plus d'informations sur les fichiers délimités, voir «Fichiers délimités», à la page 110.
Formats numériques	Spécifications relatives au formatage des nombres. Les valeurs par défaut sont la virgule (,) comme séparateur de milliers et le point (.) comme séparateur décimal.
Informations sur la colonne	Dans le cadre des fichiers délimités et à largeur fixe, l'onglet **Informations sur la colonne** fournit des détails sur chaque colonne de la table, y compris le nom, le type de données et le format applicable au type. Pour en savoir davantage, reportez-vous aux sections «Fichiers délimités», à la page 110 et «Fichiers à largeur fixe», à la page 110. Dans le cadre des fichiers XML, l'onglet **Informations sur la colonne** identifie les éléments source et la manière dont ils sont mappés aux colonnes de la table de flux de données. Pour en savoir davantage, voir «Informations sur la zone XML», à la page 112.
Clé d'événement	Pour les fichiers délimités et à largeur fixe, la clé permet d'identifier les colonnes de zone clé pour les événements multilignes. Pour plus d'informations sur les événements multilignes, voir «Evénements sur plusieurs lignes», à la page 107.
Intervalle de suppression de l'état	Cet onglet comporte plusieurs options de suppression des données d'événement. Pour en savoir plus sur l'onglet **Intervalle de suppression de l'état**, voir «Intervalle de suppression de l'état», à la page 61.

Création d'un flux de données à partir d'un fichier à plat

Un flux de données est un objet qui représente un événement. Vous pouvez utiliser un fichier à plat pour créer un flux de données.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, sélectionnez l'onglet **Plan de travail**, puis cliquez sur le bouton **Activités**.
2. Cliquez sur **Créer** puis sur **Flux de données**.
3. Sélectionnez le type source **Fichier à plat**.
4. Sélectionnez **Flux de données simple**.
5. (Facultatif pour les fichiers délimités et à largeur fixe) Identifiez un fichier exemple pour faciliter le mappage des colonnes. Ce fichier est un échantillon du fichier de données réel. Les données de ce fichier sont présentées à l'étape suivante afin de vous guider tout au long du processus de mappage des données d'événement dans la table.
6. Pour les fichiers à largeur fixe, définissez l'emplacement des colonnes de données à l'aide de la boîte de dialogue Définir les largeurs des zones.
7. Identifiez les attributs de flux de données.
8. Dans la zone **Informations sur la colonne**, définissez les informations sur la colonne en fonction du format choisi. Pour plus de détails sur le type de source, voir :
   - «Fichiers délimités», à la page 110.
   - «Fichiers à largeur fixe», à la page 110.
   - «Prise en charge des fichiers XML», à la page 111.
9. Enregistrez le flux de données avec l'état activé ; il sera ainsi prêt à recevoir des données immédiatement.

Evénements sur plusieurs lignes

Un fichier à largeur fixe ou délimité peut contenir plusieurs lignes relatives au même événement.

Par exemple, un événement "bon de commande" peut contenir une ligne pour chacun des articles de la commande. Lorsque cet événement est chargé dans la table de flux de données, le système traite chacune des lignes comme faisant partie du même événement, et non comme un nouvel événement.

Pour que les lignes soient reconnues comme contenant des données relatives au même événement, chaque ligne doit comporter des données d'identification propres à l'événement. Les données d'exemple ci-après contiennent des articles figurant sur trois bons de commandes, chacune des commandes étant identifiée par la colonne POID. La première commande comporte trois articles, la deuxième un article et la troisième deux articles :

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0697,3,Locking Rail Key,25,7.50,187.50
0698,1,Nano Webber,50,6.00,300.00
0699,1,Tall Bar Stool,100,60.00,6000.00
0699,2,Can of Levers,250,1.50,375.00
```

Lorsque vous définissez les informations de colonne de cet événement, vous identifiez la colonne POID en tant que zone clé en choisissant **Clé d'événement**. Chaque événement peut posséder une ou plusieurs zones clé d'événement.

Les lignes associées à chaque événement doivent être regroupées dans le fichier et les données des zones Clé d'événement doivent être propres à l'événement. Dès que les données de l'une des zones diffèrent de celles de la ligne précédente, la ligne correspondant à la zone fait référence à un nouvel événement. Par exemple, les données ci-après sont considérées comme trois événements distincts, même si la dernière ligne comporte la même valeur de clé d'événement que les deux premières.

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0100,1,Tall Bar Stool,100,60.00,6000.00
0697,3,Locking Rail Key,25,7.50,187.50
```

Si l'une des lignes contient des données non valides, elle est rejetée et n'a pas de répercussions sur les lignes suivantes. Par exemple, la troisième ligne de données ci-après contient un caractère (X) à un emplacement destiné à un nombre entier. Dans cet exemple, la troisième ligne est donc rejetée, et la quatrième est insérée en tant que troisième ligne de l'événement.

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0697,X,Tall Bar Stool,100,60.00,6000.00
0697,4,Locking Rail Key,25,7.50,187.50
```

Fuseaux horaires associés aux données de fichiers à plat

Pour les données provenant des fichiers à plat (délimités et à largeur fixe), vous pouvez définir un format de date pour une zone d'horodatage.

Outre le format de date, vous pouvez également spécifier un fuseau horaire. Cette opération vous permet de préciser la façon dont le serveur interprète la valeur d'horodatage dans la zone. En effet, l'horodatage peut être supposé appartenir au même fuseau horaire que le serveur d'applications ou à un fuseau horaire différent. Cette fonctionnalité peut se révéler utile lorsque le serveur d'applications est situé dans un autre fuseau horaire que celui dans lequel les événements se produisent et que vous souhaitez afficher l'horodatage dans une vue pour ce fuseau horaire. La valeur par défaut est le fuseau horaire dans lequel se trouve le serveur.

Le tableau ci-après fournit des exemples d'horodatages avec différents formats et fuseaux horaires. Notez que lorsque le fuseau horaire est défini sur GMT+5:30 et GMT+13:00, l'horodatage dans la vue résultante présente la date sous la forme 2006-10-25 au lieu de la valeur de la zone.

Tableau 48. Exemples d'horodatages

Valeur de zone dans le fichier à plat	Format de l'horodatage	Résultat de la vue
2006-10-26 09:49:12.976	yyyy-MM-dd HH:mm:ss.S ; Fuseau horaire du serveur	2006-10-26 09:49:12.976

Tableau 48. Exemples d'horodatages (suite)

Valeur de zone dans le fichier à plat	Format de l'horodatage	Résultat de la vue
2006-10-26 09:49:12.976	yyyy-MM-dd HH:mm:ss.S ; (GMT+5:30) Chennai, Kolkata, Mumbai ...	2006-10-25 21:19:12.976
2006-10-26 09:49:12	yyyy-MM-dd HH:mm:ss ; (GMT+13:00)Nuku'alofa	2006-10-25 13:49:12.0

Définition d'un format de date et d'un fuseau horaire personnalisés

Les données de fichiers à plat délimités et à largeur fixe vous offrent la possibilité de définir un format de date et un fuseau horaire personnalisés afin de préciser la façon dont le serveur interprète la valeur d'horodatage dans la zone.

Procédure

1. Cliquez sur **Modifier ce flux de données** pour le flux de données.
2. Dans l'onglet **Informations sur la colonne**, recherchez la zone contenant un type de données TIMESTAMP, puis sélectionnez **Changer la mise en forme** sous **Mise en forme**.
3. Sélectionnez un format dans le menu déroulant **Format date/heure** ou créez un format personnalisé.
4. Sélectionnez un fuseau horaire dans le menu déroulant **Fuseau horaire**.
5. Cliquez sur **OK**.

   Le format d'horodatage et le fuseau horaire s'affichent dans la colonne **Mise en forme** de votre zone d'horodatage.

Création d'un format de date personnalisé

Les données de fichiers à plat délimités et à largeur fixe vous offrent la possibilité de définir un format de date personnalisé afin de préciser la façon dont le serveur interprète la valeur d'horodatage dans la zone.

Procédure

1. Ouvrez la fenêtre Format de date personnalisé.
2. Sélectionnez **Format personnalisé** dans le menu déroulant **Format date/heure**.
3. Créez votre format personnalisé dans la zone **Format personnalisé** en entrant les lettres composant le format de date et d'heure.

   Pour connaître les définitions disponibles, utilisez la liste de formats de date proposée. La zone **Exemple** présente un exemple du format que vous créez. Ainsi, si vous créez un format personnalisé pour `E MMM/dd/yyy G h:m:s.S a 'UTC'`, la zone d'exemple affiche :

   ```
   Fri June/14/2010 AD 10:49::6.144 AM UTC
   ```
4. Cliquez sur **OK**.

Fichiers délimités

Dans un fichier délimité, chaque zone (colonne) est séparée par un caractère, une virgule par exemple.

Par exemple :

```
0703,00001,Assigned,13,2003-03-05 14:23:00,Sridar
0706,00004,Open,13,2003-03-05 19:50:00,
0706,00004,Resolved,13,2003-03-05 19:50:00,Niku
```

Les fichiers délimités sont également appelés fichiers CSV.

Le séparateur de zone, le caractère d'échappement et le qualificatif de texte sont tous personnalisables.

- Caractère de séparation

  Sépare chacune des zones de la ligne ; il s'agit le plus souvent d'une virgule (,).
- Caractère d'échappement

  Précède les caractères qui ne doivent pas être utilisés comme séparateurs ; il s'agit généralement d'une barre oblique inversée (\). Par exemple, si le caractère de séparation est une virgule et que le texte contient également une virgule, cette dernière est précédée d'une barre oblique inversée. Dans l'exemple ci-après, la virgule qui suit "Altadena" n'est pas un séparateur de zone :

  ```
  123 Buena Loma Dr,Altadena\, CA,91001
  ```
- Qualificatif de texte

  Caractère délimitant davantage les chaînes de texte ; il s'agit généralement de guillemets ("). Utilisez cette option pour des chaînes de texte qualifiées comme étant différentes des autres types de données. Par exemple, l'événement ci-après comporte des zones de texte contenant des chiffres, mais l'une de ces zones (valeur 13) est numérique.

  ```
  "0706","00004","Open",13,2010-03-05
  19:50:00,""
  ```

Les lignes source qui ne contiennent pas suffisamment de données pour remplir la ligne génèrent une erreur. Pour autoriser ces données sans obtenir de message d'erreur, activez l'option **Autoriser les lignes courtes**. Par exemple, le texte ci-après génère une erreur dès que la troisième ligne est importée, sauf si les lignes courtes sont autorisées :

```
1,2,3,4,5,6
1,,,,,
1
```

Lorsque l'option **Utiliser cette ligne pour les noms de colonne** est sélectionnée, les noms indiqués dans la ligne s'affichent comme noms de colonne. Dans le cas contraire, définissez manuellement les noms. En outre, pour chacune des colonnes, indiquez un type de données et, en option, déclarez un format de données.

Fichiers à largeur fixe

Dans les fichiers à largeur fixe, chacune des zones (colonnes) présente la même largeur prédéfinie sur chaque ligne du fichier, comme dans une table de feuille de calcul.

Par exemple :

```
070300001Assigned13  2010-03-05 14:23:00Sridar
0706000004Open    13  2010-03-05 19:50:00
070600004Resolved13  2010-03-05 19:50:00Niku
```

Pour importer un fichier à largeur fixe, utilisez la boîte de dialogue **Définir les largeurs des zones** permettant d'identifier les positions des colonnes qui commencent chaque zone de données. Lorsque vous fournissez un exemple et que les données d'exemple s'affichent, vous cliquez sur les colonnes pour indiquer le début de la zone.

Si vous ne disposez pas d'un exemple, vous devez identifier la position de départ de chacune des colonnes dans le texte. N'oubliez pas que la première zone démarre à la position zéro (0).

Une fois les positions des colonnes définies, vous pouvez leur affecter des noms et déclarer le type et le format des données qu'elles contiennent.

Lorsque l'option **Utiliser cette ligne pour les noms de colonne** est sélectionnée, les noms indiqués dans la ligne s'affichent comme noms de colonne. Dans le cas contraire, définissez manuellement les noms. En outre, pour chacune des colonnes, indiquez un type de données et, en option, déclarez un format de données.

Prise en charge des fichiers XML

IBM Cognos Real-time Monitoring prend en charge la génération d'événements à partir de sources de données au format XML.

Toutefois, les vues et les tables de flux de données de Cognos Real-time Monitoring se présentent sous la forme de tableaux (c'est-à-dire, avec des lignes et des colonnes) tandis que les données XML se présentent sous la forme hiérarchique. Ainsi, le fichier XML doit être mis à plat dans une table de flux de données alors que les données hiérarchiques sont mappées à des données tabulaires. Le processus de mise à plat des fichiers XML est décrit à la section «Procédure de mise à plat des fichiers XML», à la page 115.

Pour définir un objet de flux de données XML dans Cognos Real-time Monitoring, un concepteur d'applications doit indiquer les colonnes à afficher dans la table de flux de données. Chaque colonne de la table de flux de données possède un nom, un type et une source. Chaque colonne comporte une source spécifiée sous la forme d'une expression XPath, ce qui signifie que la colonne se remplit au fur et à mesure que des éléments correspondants à l'expression XPath donnée sont détectés, en utilisant la valeur de l'élément dans l'entrée XML en tant que valeur de la colonne dans une ligne de sortie. Chaque colonne XPath doit obligatoirement spécifier un élément ou un attribut. Généralement, une colonne XPath qui spécifie un élément correspond uniquement aux éléments atomiques qui se trouvent dans le document d'entrée, à savoir, les éléments qui contiennent uniquement du texte et aucun autre élément. Pour en savoir davantage, voir «Méthode d'évaluation des éléments», à la page 123.

Le mappage des données XML hiérarchiques à une table est possible uniquement si la source de chaque colonne de table est définie en tant qu'expression XPath. Les colonnes de la table se remplissent au fur et à mesure que des éléments correspondants à l'expression XPath sont détectés, en utilisant la valeur des éléments dans l'entrée XML et la valeur dans la colonne d'une ligne de sortie. Chaque colonne XPath spécifie un élément atomique ou un attribut. Pour plus

d'informations sur la manière dont les expressions XPath permettent de localiser les lignes et les colonnes, voir «Emplacement XPath des colonnes et des lignes dans les fichiers XML».

Emplacement XPath des colonnes et des lignes dans les fichiers XML

Vous pouvez utiliser les expressions XPath pour localiser les colonnes et les lignes d'un fichier XML.

Lorsque vous définissez la zone **Informations sur la zone** associée à un objet de flux de données, les expressions XPath localisent les colonnes et les lignes dans le fichier XML, comme suit :

- Le schéma XPath identifie l'expression XPath absolue de l'élément du document dans lequel résident toutes les autres données XML définies par les expressions XPath. Il peut s'agir d'une barre oblique (/) faisant référence au noeud racine du document XML, ou d'une expression XPath plus complexe. Si un document XML comporte plusieurs éléments qui satisfont le schéma XPath, une table est générée pour cet élément et la sortie du document est définie en tant qu'union de la sortie des éléments correspondant au schéma XPath. Un schéma XPath est un chemin d'accès absolu à l'élément dans la structure XML et commence toujours par une barre oblique (/), suivie de l'élément racine et du chemin à l'élément de colonne, par exemple "/problem_tickets/ticket/ticket_id".
- Les expressions XPath relatives identifient un élément de colonne ou un attribut concernant l'élément de ligne. Lorsque la colonne est un élément enfant de l'élément de ligne, l'expression XPath correspond uniquement au nom de l'élément ou commence par "child::". Vous trouverez ci-dessous des expressions XPath relatives valides issues de l'exemple précédent :

  ```
  ticket_id
    child::status
    customer/customer_name
    child::customer/customer_name
  ```

  Pour localiser un attribut, placez un symbole @ avant le nom de l'attribut :

  ```
    customer/@cust_id
  ```

L'expression XPath standard pour les emplacements définit des mappages de noeud XML supplémentaires non pris en charge par les tables de flux de données IBM Cognos Real-time Monitoring.

Informations sur la zone XML

Chaque colonne de la table de flux de données est définie en tant que zone dans l'éditeur de flux de données XML.

Chaque zone possède les attributs décrits dans le tableau ci-dessous :

Tableau 49. Attributs de la zone XML

Attribut	Description
Xpath de schéma	L'expression XPath de schéma sélectionne une liste de noeuds dans laquelle chaque noeud encapsule une ligne dans l'ensemble de données final. Pour plus d'informations sur les expressions XPath de schéma, voir «Emplacement XPath des colonnes et des lignes dans les fichiers XML», à la page 112.
Nom de zone	Nom de la colonne dans la table de flux de données.
XPATH relatif	Elément du fichier XML qui comporte les données de la zone. Pour plus d'informations sur les expressions XPath, voir «Emplacement XPath des colonnes et des lignes dans les fichiers XML», à la page 112.
Type de données XML	Type de données associé à l'élément XML. Pour plus d'informations sur les types de données XML, voir «Types de données XML».
Type de données	Type de données IBM Cognos Real-time Monitoring associé à la colonne de la table de flux de données. Pour en savoir davantage, voir Chapitre 9, «Types de données», à la page 69.
Mise en forme	Formatage de la valeur décimale, chaîne ou date-heure.

Types de données XML

Mappage des types de données XML aux types de données IBM Cognos Real-time Monitoring.

Le mappage est décrit dans le tableau ci-dessous.

Tableau 50. Table de mappage

XSD	Type de données
anyURI	Varchar
base64Binary	Varchar
Booléen	Booléen
byte	Entier
date	Horodatage (portion d'heure mise à zéro)
dateTime	Horodatage (notez que l'application prend en charge jusqu'à 9 chiffres significatifs pour la partie fractionnaire d'une seconde)

Tableau 50. Table de mappage (suite)

XSD	Type de données
decimal	Décimal si l'échelle n'est pas égale à 0. Si l'échelle est égale à 0, le type de données est déterminé par la précision, comme suit : Entier si la précision est comprise entre 1 et 9. Long si la précision est comprise entre 10 et 18. Décimal si la précision est supérieure à 18.
double	Double
duration	Varchar (sous forme de chaîne)
ENTITIES	Varchar
ENTITY	Varchar
float	Double
gDay	Varchar (sous forme de chaîne) définissant le jour (DD)
gMonth	Varchar (sous forme de chaîne) définissant le mois (MM) au sein de la date
gMonthDay	Varchar (sous forme de chaîne) définissant le mois et le jour (MM-DD) au sein de la date
gYear	Varchar (sous forme de chaîne) définissant l'année (CCYY) au sein de la date
gYearMonth	Varchar (sous forme de chaîne) définissant l'année et le mois (CCYY-MM) au sein de la date
hexBinary	Varchar
ID	Varchar
IDREF	Varchar
IDREFS	Varchar
int	Entier
integer	Entier
language	Varchar
long	Long
Nom	Varchar
NCName	Varchar
negativeInteger	Entier
NMTOKEN	Varchar
NMTOKENS	Varchar (sous forme de chaîne)
nonNegativeInteger	Entier
nonPositiveInteger	Entier
normalizedString	Varchar
NOTATION	Varchar
positiveInteger	Entier
QName	Varchar
QName	Varchar

Tableau 50. Table de mappage (suite)

XSD	Type de données
short	Entier
string	Varchar
time	Varchar
token	Varchar
unsignedByte	Entier
unsignedInt	Décimal
unsignedLong	Décimal
unsignedShort	Entier

Procédure de mise à plat des fichiers XML

Le mappage des données XML hiérarchiques à une table est possible uniquement si la source de chaque colonne de table est définie en tant qu'expression XPath.

Les exemples de cette section fournissent toujours l'expression XPath absolue pour une colonne. Toutefois, lorsque vous configurez un objet de flux de données XML, vous utilisez une expression XPath de schéma absolue et les événements de colonne correspondent aux expressions XPath associées à l'expression XPath de schéma.

Les colonnes de la table se remplissent au fur et à mesure que des éléments correspondants à l'expression XPath donnée sont détectés, en utilisant la valeur des éléments dans l'entrée XML et la valeur dans la colonne d'une ligne de sortie. Par exemple, le document XML suivant possède les expressions XPath : a/b, a/c, a/d :

```
<a>1
    <b>0</b>
    <c>1</c>
    <d>2</d>
</a>
```

La table résultat comporte une seule ligne :

0, 1, 2

Dans l'exemple suivant, le document est légèrement plus compliqué et comporte les expressions XPath : a/b, a/c/d a/c/e/f.

```
<a>
   <b>0</b>
   <c>
     <d>1<d>
      <e>
          <f>2</f>
      </e>
     <d>
   </c>
</a>
```

La table résultat comporte également une seule ligne :

0,1,2

Dans ces deux exemples, les documents XML apparaissent sous la forme d'une table à une seule ligne, car chaque expression XPath correspond exactement à un élément. La table résultat comporte une ligne, dans laquelle chaque colonne reçoit la valeur de l'élément qui correspond à l'expression XPath qui définit la source pour une colonne.

Dans certains cas, plusieurs éléments peuvent correspondre à une expression XPath pour une colonne source. Dans l'exemple suivant, les expressions XPath sont a/b et a/c, qui correspondent à plusieurs colonnes.

```
<a>
   <b>0</b>
   <b>1</b>
   <c>2</c>
   <d>3</d>
</a>
```

La table résultat devrait apparaître sous la forme suivante :

```
0,2
0,3
```

Toutefois, elle se présente comme suit :

```
0,2
0,3
1,2
1,3
```

Les quatre éléments sont les enfants d'un parent unique <a> et la source des colonnes pour chaque ligne est définie par les expressions XPath a/b et a/c. Autrement dit, les lignes 0,2 et 0,3 ne sont plus correctes et devraient se trouver dans la table. Dans ce cas, la table résultat est créée à partir d'un produit croisé d'éléments.

Pour garantir l'exactitude du résultat lors de la mise à plat des fichiers XML, IBM Cognos Real-time Monitoring effectue les opérations suivantes :

- Supprime le fichier XML d'entrée de tous les éléments externes de sorte que les valeurs ne soient pas dupliquées lorsque cela est inutile, ce qui aurait pour conséquence de générer des lignes en double dans la sortie.
- Structure le document qui est vidé de ses éléments externes.

Pour accomplir cette dernière tâche, le document est réparti dans plusieurs tables intermédiaires, dans lesquelles chaque table possède une clé primaire (pk), une clé externe (parent_fk) et une valeur. La table résultat finale est créée via une jointure de clé de l'ensemble des tables intermédiaires. Par exemple, le document XML suivant peut être structuré en huit tables intermédiaires a, a/b, a/b/c, a/b/d, a/b/b/d/e, a/b/d/f, a/b/d/f/g et a/h.

```
<a>
   <b>
      <c>0</c>
      <d>
         <e>1</e>
```

```
        <e>2</e>
           <f>
               <g>3</g>
           </f>
      </d>
    </b>
    <b>
       <c>4</c>
       <d>
          <e>5</e>
          <f>
              <g>6</g>
              <g>7</g>
          <f/>
        </d>
   </b>
   <h>8</h>
   <h>9</h>
</a>
```

Les clés primaires (pk) sont ensuite affectées à chaque élément lors d'un parcours en profondeur d'abord du document.

```
<a pk='0'>
   <b pk='1'>
      <c pk='2'>0</c>
      <d pk='3'>
         <e pk='4'>1</e>
         <e pk='5'>2</e>
            <f pk='6'>
                <g pk='7'>3</g>
            </f>
      </d>
    </b>
    <b pk='8'>
       <c pk='9'> 4 </c>
       <d pk='10'>
          <e pk='11'> 5 </e>
          <f pk='12'>
              <g pk='13'>6</g>
              <g pk='14'>7</g>
          <f/>
        </d>
   </b>
   <h pk='15'>8</h>
   <h pk='16'>9</h>
</a>
```

A présent, les clés des tables sont jointes à l'aide des clés primaires affectées à chaque élément. L'instruction SELECT suivante définit la table de sortie finale pour la table de flux de données :

```
SELECT a/b/c.value, a/b/d/e.value, a/b/d/f/g.value, a/h.value
FROM ((((((a/
     FULL OUTER JOIN /a/b on /a/b.parent_fk = /a.pk)
     FULL OUTER JOIN /a/b/c on /a/b/c.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d on /a/b/d.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d/e on /a/b/d/e.parent_fk =
/a/b/d.pk)
     FULL OUTER JOIN /a/b/d/f on /a/b/d/f.parent_fk =
/a/b/d.pk)
     FULL OUTER JOIN /a/b/d/f/g on /a/b/d/f/g.parent_fk
= /a/b/d/f.pk)
     FULL OUTER JOIN /a/h on /a/h.parent_fk = a.pk
```

La même jointure peut être réalisée à l'aide de la clé primaire de chaque table :

```
SELECT /a.pk, /a/b.pk, /a/b/c.pk, /a/b/d.pk, /a/b/d/e.pk,
/a/b/d/f.pk, /a/b/d/f/g.pk, /a/h.pk
FROM ((((((/a
     FULL OUTER JOIN /a/b on /a/b.parent_fk = /a.pk)
     FULL OUTER JOIN /a/b/c/ on /a/b/c.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d on /a/b/d.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d/e on /a/b/d/e.parent_fk =
/a/b/d.pk)
     FULL OUTER JOIN /a/b/d/f on /a/b/d/f.parent_fk =
/a/b/d.pk)
     FULL OUTER JOIN /a/b/d/f/g on /a/b/d/f/g.parent_fk
= /a/b/d/f.pk)
     FULL OUTER JOIN /a/h on /a/h.parent_fk = a.pk
```

La table résultat est la suivante :

```
0,1,3,8
0,2,3,8
4,5,6,8
4,5,7,8
0,1,3,9
0,2,3,9
4,5,6,9
4,5,7,9
```

Mode de traitement des éléments manquants

Dans le cas où les éléments ne possèdent pas les mêmes éléments enfant, une jointure externe complète est utilisée pour générer la table, car la définition d'une sortie avec une jointure interne ne produirait pas les résultats attendus.

Par exemple, le document suivant ne comporte pas d'élément enfant <d> dans le premier élément <b>, tandis que le second élément <b> comporte un élément <d> mais pas d'élément <c> :

```
<a>
   <b>
      <c>0</c>
```

```
   </b>
   <b>
      <d>1</d>
   </b>
</a>
```

Le document annoté ressemble à ce qui suit :

```
<a pk='0'>
   <b pk='1'>
      <c pk='2'>0</c>
   </b>
   <b>
      <d pk='3'>1</d>
   </b>
</a>
```

L'instruction SELECT se présente comme suit :

```
SELECT /a/b/c.value, /a/b/d.value
FROM ((/a
     FULL OUTER JOIN /a/b on /a/b.parent_fk = /a.pk)
     FULL OUTER JOIN /a/b/c on /a/b/c.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d on /a/b/d.parent_fk = /a/b.pk
```

L'instruction SELECT avec projection de clé primaire se présente comme suit :

```
SELECT /a.pk, /a/b.pk, /a/b/c.pk, /a/b/d.pk
FROM ((/a
     FULL OUTER JOIN /a/b on /a/b.parent_fk = /a.pk)
     FULL OUTER JOIN /a/b/c on /a/b/c.parent_fk = /a/b.pk)
     FULL OUTER JOIN /a/b/d on /a/b/d.parent_fk = /a/b.pk
```

Le résultat final est le suivant :

```
0,null
null,1
```

Mode de traitement des attributs

Lorsque les éléments d'un document contiennent des attributs, ces derniers sont représentés sous la forme de colonnes dans la table intermédiaire.

Par exemple, dans le document suivant, les expressions XPath de la colonne sont les suivantes : a/@b, a/c/@d et a/c.

```
<a b='0'>
   <c d='1'>2</c>
   <c d='3'>4</c>
</a>
```

Les clés primaires sont affectées aux éléments de la manière suivante :

```
<a pk='0' b='0'>
   <c pk='1' d='1'> 2</c>
   <c pk='2' d='3'> 4 </c>
</a>
```

L'instruction SELECT se présente comme suit :

```
SELECT /a/@b, /a/c.d, /a/c.value
FROM /a FULL OUTER JOIN /a/c on /a/c.parent_fk = /a.pk
```

L'instruction SELECT avec clés primaires se présente comme suit :

```
SELECT /a/@b, /a/c.d, /a/c.pk
FROM /a FULL OUTER JOIN /a/c on /a/c.parent_fk = /a.pk
```

La table résultat est la suivante :

```
0,1,2
0,3,4
```

Lorsqu'un élément XML comporte deux enfants dotés du même nom de balise, une simple expression XPath sans prédicat renvoie un ensemble de noeuds. Par exemple, dans le document suivant, l'expression XPath /a/b renvoie deux lignes :

```
<a>
   <b>0</b>
   <b>1</b>
</a>
```

Les lignes renvoyées sont les suivantes :

```
0
1
```

Dans les expressions XPath, vous pouvez utiliser des prédicats d'index pour rechercher un noeud ou une valeur spécifique dans un noeud. Par exemple, l'expression /a/b[1] renvoie 0 et l'expression /a/b[2] renvoie 1. Autrement dit, chaque expression XPath renvoie une ligne. L'expression /a/b[3] ne renvoie rien, car il n'y a pas de troisième enfant dans l'élément <a>. Dans l'exemple ci-dessous, l'expression XPath /a/b/c[1] renvoie deux lignes.

```
<a>
   <b>
      <c>0</c>
      <c>1</c>
   </b>
   <b>
      <c>2</c>
      <c>3</c>
   </b>
</a>
```

Les lignes renvoyées sont les suivantes :

```
0
2
```

Cela est dû au fait que l'expression XPath /a/b/c[1] indique le premier enfant dans l'élément <b> et qu'il y a deux éléments enfant <b> dans l'élément parent <a>.

Eléments ignorés

Les éléments d'un document qui ne correspondent à aucune expression XPath de colonne ou qui ne possèdent aucun enfant associé à une expression XPath de colonne sont ignorés.

Dans l'exemple suivant, le document comporte l'expression XPath /a/b/c, mais une seule ligne est renvoyée.

```
<a>
   <b>
      <c>0</c>
      <d>
         <e>1</e>
         <e>2</e>
      </d>
   </b>
<a>
```

En effet, l'élément /a/b/d ne correspond pas à l'expression XPath /a/b/c de la colonne et il ne possède aucun enfant associé. Par conséquent, à l'issue de la première transmission, le document aura l'aspect suivant :

```
<a>
   <b>
      <c>0</c>
   </b>
<a>
```

Le document suivant génère également une seule ligne.

```
<a>
   <b>
      <c>0</c>
   </b>
   <b>
      <d>1</d>
   </b>
</a>
```

En effet, le deuxième élément /ab correspond à l'expression XPath de la colonne. Toutefois, il ne possède aucun enfant associé. Par conséquent, à l'issue de la première transmission, le document aura également l'aspect suivant :

```
<a>
   <b>
      <c>0</c>
   </b>
<a>
```

Le même processus se produit avec les attributs. Dans l'exemple suivant, le document comporte l'expression XPath /a/b/@id et il ne génère qu'une seule ligne à l'issue de la première transmission, car le deuxième élément <b> n'est associé à aucun attribut id.

```
<a>
   <b id='0' name='oracle'/>
```

```
   <b name='ibm'/>
</a>
```

Après la première transmission, le document ressemble à ce qui suit :

```
<a>
   <b id='0' name='oracle'/>
</a>
```

Prédicats d'index

Les prédicats d'index permettent de surmonter un problème particulier, à savoir qu'une ligne doit être renvoyée, mais que le code XML contient deux balises enfant portant le même nom, mais avec un contenu différent.

Par exemple, le document suivant contient deux occurrences de la balise <keyword> :

```
<sales>
   <sale>
      <id>0</id>
      <amount>$125</amount>
      <keyword>Salesforce </keyword>
      <keyword>SAP</keyword>
   </sale>
</sales>
```

Pour l'expression XPath de schéma /sales, les expressions XPath de colonne sont les suivantes :

```
sale/id
sale/amount
sale/keyword
```

Deux lignes sont alors renvoyées :

```
0, $125, Salesforce
0, $124, SAP
```

Toutefois, une seule vente n'a eu lieu. Par conséquent, une seule ligne aurait dû être renvoyée : 0, $125, Salesforce. En d'autres termes, vous vouliez créer une vue au-dessus de l'événement de ventes SELECT SUM(amount) FROM et vous n'avez besoin que d'un mot clé pour cela. Dans ce cas, vous pouvez indiquer les expressions XPath suivantes :

```
sale/id
sale/amount
sale/keyword[1]
```

La résultat serait la ligne : 0, $124, Salesforce. Cependant, si vous avez besoin des deux mots clés, vous devez utiliser une méthode différente de celle des prédicats d'index.

Les prédicats d'index peuvent exister à n'importe quel niveau d'une expression XPath, et pas seulement au dernier niveau. Par exemple, l'expression /a/b[2]/c renvoie deux lignes basées sur le document suivant :

```
<a>
   <b>
```

```
      <c>0</c>
      <c>1</c>
   </b>
   <b>
      <c>2</c>
      <c>3</c>
   </b>
</a>
```

Ces deux lignes sont les suivantes :

```
2
3
```

Méthode d'évaluation des éléments

Dans certains cas, les éléments renvoient une valeur NULL lorsqu'ils sont évalués par IBM Cognos Real-time Monitoring.

Cela se produit dans les cas suivants :

- L'élément est vide. Par exemple, il existe un élément /a/b dans le document suivant, que Cognos Real-time Monitoring définit comme NULL :

  ```
  <a>
     <b></b>
  </a>
  ```

- L'élément possède des enfants mais pas de texte. Par exemple, le document suivant contient un élément <c>, qui possède un élément enfant <d> mais qui ne contient aucun texte.

  ```
  <a>
     <b>0</b>
        <c>
           <d>1</d>
        </c>
        <c>2</c>
  </a>
  ```

  La sortie du document, conformément aux expressions XPath /a/b, a/b/c et /a/b/c/d est la suivante :

  ```
  0, null, 1
  0, 2, null
  ```

  Si un élément n'est pas un élément atomique et qu'il contient un élément enfant, il renvoie le texte de l'élément ainsi que de l'enfant. Par exemple, le document suivant contient l'élément <b>, qui n'est pas atomique, et un élément enfant <c>:

  ```
  <a>
     <b>hello
        <c>world</c>
     com</b>
  </a>
  ```

  Les colonnes XPath /a/b et /a/b/c renvoient les lignes suivantes :

  ```
  hello
     , com, world
  ```

Problèmes liés à la mise à plat des fichiers XML

Tenez compte des problèmes ci-dessous lors de la mise à plat de fichiers XML.

Il est incorrect de mélanger les prédicats. Par exemple, l'ensemble d'expressions XPath /a/b[2], /a/b est incorrect. Ceci est dû au fait que, selon la sémantique utilisée par IBM Cognos Real-time Monitoring pour mettre à plat le fichier XML, dès que les prédicats d'index sont utilisés avec un préfixe XPath donné, tous les éléments correspondants à ce préfixe sont implicitement réécrits en ajoutant l'index. Chaque élément doit être écrit avec ou sans le prédicat d'index. Lorsque vous indiquez un préfixe XPath donné dans une colonne avec un prédicat d'index et dans une autre colonne sans prédicat d'index, tous les éléments correspondants à ce préfixe sont ambigus.

Lorsque les données sont issues d'un fichier à plat, chaque correspondance au schéma XPath définit un nouvel événement, qui peut définir un nombre illimité de lignes. Par exemple, un événement d'un fichier à plat comportant le schéma XPath /a/b et la colonne XPath c génère deux événements du document ci-dessous, chacun avec une seule ligne :

```
<a>
   <b>
      <c>0</c>
   </b>
   <b>
      <c>1</>
   </b>
</a>
```

Cependant, lorsqu'un document XML provient d'un service JMS, un document génère toujours un événement. (Les différences sont rares.)

Les caractères vides sont incorrects dans les expressions XPath. L'enregistrement d'un événement qui comporte des espaces blancs dans les expressions XPath est impossible. Les expressions XPath suivants sont incorrectes :

- b/ c
- b/c [1]

Si vous indiquez plusieurs colonnes dans la même expression XPath, vous obtenez toujours la même valeur dans les deux colonnes. Par exemple, le document ci-dessous renvoie le résultat 0,1,1 compte tenu du schéma /a et les colonnes b, c, c.

```
<a>
   <b>0</b>
   <c>1</b>
<a>
```

La requête sémantique est la suivante :

```
SELECT /a/b.value, /a/c.value, /a/c.value
FROM
   (/a FULL OUTER JOIN /a/b ON /a/b.parent_fk = /a.pk)
   FULL OUTER JOIN /a/c ON /a/c.parent_fk = /a.pk
```

Chapitre 14. Formules

Toutes les formules d'IBM Cognos Real-time Monitoring sont des expressions formulées en langage C-SQL, un dérivé du langage ANSI SQL.

Certaines de ces formules sont des expressions simples, telles que des expressions de zone définissant les valeurs des colonnes d'une vue métier. D'autres expressions plus complexes représentent des composants entiers de l'instruction de requête C-SQL (SELECT), comme les clauses WHERE, WINDOW et JOIN.

Toutes les formules d'IBM Cognos Real-time Monitoring acceptent des opérateurs et des constantes (voir Chapitre 22, «Opérateurs et constantes», à la page 291) pouvant manipuler les valeurs, ainsi que la plupart des fonctions C-SQL (voir Chapitre 15, «Fonctions C-SQL», à la page 133) permettant de traiter plus avant les résultats.

Pour obtenir une description détaillée de l'instruction SELECT, des opérateurs et des fonctions C-SQL, voir Chapitre 32, «Instructions SELECT», à la page 337.

Fonctions

Les fonctions renvoient des valeurs qui sont des informations système, telles que l'heure en cours, des manipulations de données, telles que la conversion d'une chaîne de caractères en lettres majuscules, ou des évaluations d'ensembles de données, telles que le montant total d'un groupe de bons de commande.

Les fonctions C-SQL sont utilisables dans la plupart des formules d' IBM Cognos Real-time Monitoring. Toutefois, certaines de ces fonctions sont limitées par les opérations autorisées dans le contexte de la formule. La section suivante "«Types de fonction»", décrit les types de fonctions et indique les cas dans lesquels vous pouvez les utiliser.

Si vous recherchez une fonction pour une tâche spécifique, voir «Catégories de fonctions», à la page 128 pour connaître les tâches que les fonctions vous permettent d'effectuer.

Pour plus de détails sur chacune des fonctions, voir Chapitre 15, «Fonctions C-SQL», à la page 133.

Types de fonction

C-SQL comporte cinq types de fonctions : scalaire (Scalar), ensemble (Set), classement (Rank), ensemble mobile (Moving set) et ensemble en défilement continu (Tumbling set). Le type de fonction détermine les cas dans lesquels vous êtes autorisé à utiliser la fonction dans une formule.

Fonctions scalaires

Les fonctions scalaires (Scalar) agissent sur un seul élément et renvoient un résultat unique.

© Copyright IBM Corp. 2007, 2014

Par exemple, la fonction ABS (voir «ABS», à la page 152) renvoie la valeur absolue d'un nombre (unique). Les fonctions scalaires peuvent apparaître dans n'importe quelle expression C-SQL.

Ces fonctions sont décrites dans les sections suivantes :

- «ABS», à la page 152
- «CAST», à la page 142
- «CEIL», à la page 152
- «CHAR_LENGTH», à la page 167
- «CONCAT», à la page 168
- «CURRENT_TIMESTAMP», à la page 144
- «CURRENT_USER», à la page 166
- «DATE_ADD», à la page 144
- «DATE_DIFF», à la page 145
- «DISPLAY_MONEY», à la page 153
- «EXP», à la page 155
- «FLOOR», à la page 155
- «GREATEST», à la page 146
- «IS_RAISED», à la page 165
- «LAST_DAY», à la page 147
- «LEAST», à la page 147
- «LOG», à la page 157
- «LOWER», à la page 170
- «LPAD», à la page 170
- «LTRIM», à la page 171
- «MOD», à la page 158
- «POSITION», à la page 172
- «POWER», à la page 159
- «PRIOR_VALUE», à la page 148
- «ROUND», à la page 160
- «RPAD», à la page 173
- «RTRIM», à la page 174
- «SIGN», à la page 162
- «SQRT», à la page 163
- «SUBSTRING», à la page 175
- «TIMESTAMP_DIFF», à la page 149
- «TO_CHAR», à la page 150
- «TO_DATE», à la page 151
- «TRUNC», à la page 163

Fonctions d'ensembles

Les fonctions d'ensembles (Set) agrègent des ensembles de lignes d'une vue métier et génèrent un résultat unique pour ces ensembles.

Par exemple, la fonction SUM() renvoie le total de toutes les lignes d'une colonne figurant dans une vue. Une fonction d'ensembles est uniquement utilisable dans la Liste Select d'une instruction SELECT : définitions de zone d'une vue.

Une fonction d'ensembles peut faire référence à une autre fonction d'ensembles, mais les résultats obtenus sont les mêmes que si la fonction référencée (interne) était exprimée seule. Par exemple, SUM(AVG(Order_Total)) renvoie le même résultat que AVG(Order_Total).

Les fonctions d'ensembles sont décrites dans les sections suivantes :

- «AVG», à la page 177
- «COUNT», à la page 178
- «CURRENT», à la page 179
- «MAX», à la page 179
- «MIN», à la page 180
- «PREV», à la page 187
- «SUM», à la page 188
- «STD_DEVIATION», à la page 202
- «VARIANCE», à la page 207

La valeur NULL est ignorée lors du calcul de valeurs à l'aide de la fonction d'ensembles (Set), de la fonction d'ensembles mobiles (Moving set) et de la fonction de classement (Rank). Par exemple, la moyenne de (3, NULL, 3) est 3, pas NULL ni 2.

Fonctions de classement

Les fonctions de classement (Rank) calculent le résultat scalaire d'une colonne dans chaque ligne d'un ensemble par rapport à la totalité de l'ensemble.

Une fonction de classement peut être utilisée uniquement dans la liste de sélection d'une instruction SELECT.

Ces fonctions sont décrites dans les sections suivantes :

- «NTILE», à la page 198
- «RANK», à la page 200
- «RATIO_TO_REPORT», à la page 201

Fonctions d'ensembles mobiles

Les fonctions d'ensembles mobiles (Moving set) effectuent des calculs sur un ensemble de lignes les plus récentes d'une vue.

L'ensemble de lignes à inclure est déterminé uniquement à l'arrivée d'un nouvel événement. A ce moment là, seules les dernières lignes qui répondent aux critères de l'ensemble sont incluses dans le calcul.

Les fonctions d'ensembles mobiles sont définies en appliquant la syntaxe "MOV_" à une fonction d'ensembles existante. Par exemple, pour calculer une moyenne mobile, indiquez MOV_AVG(). Un ensemble mobile peut être déterminé par un nombre d'événements ou par une durée. L'exemple suivant calcule la moyenne arithmétique de l'élément Order_Total au cours des douze dernières heures. Lorsque de nouvelles commandes arrivent dans la vue, elles sont incluses dans le calcul. Toutefois, les commandes antérieures à 12 heures sont exclues.

```
SELECT MOV_AVG(Order_Total, HOUR, 12) FROM Purchase_Orders
```

Les fonctions d'ensembles mobiles sont une manière raccourcie d'exprimer une fenêtre de requête simple. Pour obtenir une description complète, voir «MOV_function», à la page 136.

Les fonctions d'ensembles mobiles sont décrites dans les sections suivantes :

- «MOV_AVG», à la page 181
- «MOV_COUNT», à la page 182
- «MOV_MAX», à la page 183
- «MOV_MIN», à la page 184
- «MOV_SUM», à la page 186
- «MOV_STD_DEVIATION», à la page 196
- «MOV_VARIANCE», à la page 197

Fonctions d'ensembles en défilement continu

Les fonctions d'ensembles en défilement continu (Tumbling set) effectuent des calculs sur un ensemble de fenêtres des lignes d'une vue.

L'ensemble de lignes à inclure est uniquement déterminé à l'arrivée d'un nouvel événement et se vide lorsqu'il est plein. Les fonctions d'ensembles en défilement continu (Tumbling set) constituent un moyen rapide d'exprimer une requête de fenêtre en défilement continu.

Pour en savoir davantage, voir «Fenêtres en défilement continu», à la page 406.

Ces fonctions sont décrites dans les sections suivantes :

- «TUMBLE_AVG», à la page 189
- «TUMBLE_COUNT», à la page 190
- «TUMBLE_MAX», à la page 191
- «TUMBLE_MIN», à la page 193
- «TUMBLE_SUM», à la page 194
- «TUMBLE_STD_DEVIATION», à la page 204
- «TUMBLE_VARIANCE», à la page 206

Catégories de fonctions

Cette section présente les catégories de fonctions C-SQL.

Alertes

- «IS_RAISED», à la page 165

Conversion

- «CAST», à la page 142
- «DISPLAY_MONEY», à la page 153
- «TO_CHAR», à la page 150
- «TO_DATE», à la page 151

Date et heure

- «CURRENT_TIMESTAMP», à la page 144
- «DATE_ADD», à la page 144
- «DATE_DIFF», à la page 145

- «GREATEST», à la page 146
- «TIMESTAMP_DIFF», à la page 149
- «LEAST», à la page 147
- «LAST_DAY», à la page 147
- «TO_CHAR», à la page 150
- «TO_DATE», à la page 151

Mathématiques

- «ABS», à la page 152
- «CAST», à la page 142
- «CEIL», à la page 152
- «EXP», à la page 155
- «FLOOR», à la page 155
- «LOG», à la page 157
- «MOD», à la page 158
- «POWER», à la page 159
- «ROUND», à la page 160
- «SIGN», à la page 162
- «SQRT», à la page 163
- «SUM», à la page 188
- «TRUNC», à la page 163

Fonctions mathématiques

Cette section présente les fonctions mathématiques C-SQL.

- «ABS», à la page 152
- «CAST», à la page 142
- «CEIL», à la page 152
- «EXP», à la page 155
- «FLOOR», à la page 155
- «LOG», à la page 157
- «MOD», à la page 158
- «POWER», à la page 159
- «ROUND», à la page 160
- «SIGN», à la page 162
- «SQRT», à la page 163
- «SUM», à la page 188
- «TRUNC», à la page 163

Fonctions de classement

Cette section présente les différentes fonctions de classement (Rank) C-SQL.

- «NTILE», à la page 198
- «RANK», à la page 200
- «RATIO_TO_REPORT», à la page 201

Fonctions de règles

Cette section présente les fonctions de règle C-SQL.

- «IS_RAISED», à la page 165
- «CURRENT_USER», à la page 166

Fonctions statistiques

Cette section présente les fonctions statistiques C-SQL.

- «AVG», à la page 177
- «COUNT», à la page 178
- «GREATEST», à la page 146
- «LEAST», à la page 147
- «MAX», à la page 179
- «MIN», à la page 180
- «MOV_AVG», à la page 181
- «MOV_COUNT», à la page 182
- «MOV_MAX», à la page 183
- «MOV_MIN», à la page 184
- «MOV_SUM», à la page 186
- «MOV_STD_DEVIATION», à la page 196
- «MOV_VARIANCE», à la page 197
- «NTILE», à la page 198
- «RANK», à la page 200
- «RATIO_TO_REPORT», à la page 201
- «STD_DEVIATION», à la page 202
- «TUMBLE_AVG», à la page 189
- «TUMBLE_COUNT», à la page 190
- «TUMBLE_MAX», à la page 191
- «TUMBLE_MIN», à la page 193
- «TUMBLE_SUM», à la page 194
- «TUMBLE_STD_DEVIATION», à la page 204
- «TUMBLE_VARIANCE», à la page 206
- «VARIANCE», à la page 207

Fonctions Texte et Chaîne

Cette section présente les fonctions de texte et de chaîne C-SQL.

- «CAST», à la page 142
- «CHAR_LENGTH», à la page 167
- «CONCAT», à la page 168
- «DISPLAY_MONEY», à la page 153
- «GREATEST», à la page 146
- «LEAST», à la page 147
- «LOWER», à la page 170
- «LTRIM», à la page 171
- «LPAD», à la page 170
- «POSITION», à la page 172

- «RPAD», à la page 173
- «RTRIM», à la page 174
- «SUBSTRING», à la page 175
- «TO_CHAR», à la page 150
- «TO_DATE», à la page 151
- «UPPER», à la page 176

Fonctions Séries chronologiques et agrégation

Cette section présente les fonctions chronologiques et d'agrégation C-SQL.

- «AVG», à la page 177
- «COUNT», à la page 178
- «MIN», à la page 180
- «MAX», à la page 179
- «MOV_function», à la page 136
- «MOV_AVG», à la page 181
- «MOV_COUNT», à la page 182
- «MOV_MAX», à la page 183
- «MOV_MIN», à la page 184
- «MOV_SUM», à la page 186
- «MOV_STD_DEVIATION», à la page 196
- «MOV_VARIANCE», à la page 197
- «VARIANCE», à la page 207

Fonctions des vues

Cette section présente les fonctions de vues C-SQL.

- «CURRENT», à la page 179
- «IS_RAISED», à la page 165
- «PREV», à la page 187
- «PRIOR_VALUE», à la page 148

Chapitre 15. Fonctions C-SQL

Les tableaux ci-dessous fournissent une description générale des fonctions C-SQL. Vous pouvez cliquer sur les liens pour obtenir une description détaillée.

Les fonctions C-SQL sont susceptibles d'apparaître dans les commandes et dans les formules de règle acceptant une expression.

Pour obtenir des informations d'ordre général sur les fonctions, ainsi qu'une liste des catégories de fonctions, voir «Catégories de fonctions», à la page 128 et Chapitre 14, «Formules», à la page 125.

Les tableaux suivants fournissent une brève description des fonctions C-SQL, ainsi qu'un lien vers la section qui contient davantage d'informations.

Tableau 51. Fonctions C-SQL A à I

Fonction C-SQL avec lien	Brève description
«ABS», à la page 152	Renvoie la valeur absolue d'un nombre
«AVG», à la page 177	Renvoie la valeur moyenne (moyenne arithmétique) d'un ensemble de valeurs numériques
«CASE», à la page 141	Renvoie le résultat d'une expression qui correspond à une condition de concordance true
«CAST», à la page 142	Convertit une valeur d'un type Cognos Real-time Monitoring en un autre type Cognos Real-time Monitoring
«CEIL», à la page 152	Renvoie le plus petit nombre entier, arrondi vers zéro, supérieur ou égal à un nombre
«CHAR_LENGTH», à la page 167	Renvoie la longueur d'une chaîne
«CONCAT», à la page 168	Renvoie une chaîne qui est la concaténation de deux caractères ou chaînes
«concatList», à la page 208	Renvoie une chaîne correspondant à la concaténation de deux caractères ou chaînes
«concatSet», à la page 209	Renvoie un ensemble de chaînes trié dans l'ordre alphabétique
«COUNT», à la page 178	Renvoie le nombre de lignes dans une vue ou un ensemble
«CURRENT», à la page 179	Renvoie une valeur de la ligne la plus récente ou de la dernière ligne d'un ensemble
«CURRENT_TIMESTAMP», à la page 144	Renvoie la date et l'heure en cours dans le fuseau horaire du serveur
«CURRENT_USER», à la page 166	Renvoie le nom de connexion de l'utilisateur en cours
«DATE_ADD», à la page 144	Ajoute une durée à une valeur date-heure
«DATE_DIFF», à la page 145	Soustrait une durée d'une valeur date-heure
«DISPLAY_MONEY», à la page 153	Formate un nombre en valeur monétaire

© Copyright IBM Corp. 2007, 2014

Tableau 51. Fonctions C-SQL A à I (suite)

Fonction C-SQL avec lien	Brève description
«EXP», à la page 155	Renvoie e élevé à une puissance spécifique
«FLOOR», à la page 155	Renvoie le plus grand nombre entier inférieur ou égal à une expression
«gammaDist», à la page 210	Renvoie la distribution gamma d'une valeur
«GREATEST», à la page 146	Renvoie le résultat le plus élevé d'une liste de résultats d'expressions
«IS_ACKED», à la page 164	Renvoie Vrai lorsque l'alerte est reçue
«IS_RAISED», à la page 165	Renvoie vrai lorsque l'alerte spécifiée est déclenchée

Tableau 52. Fonctions C-SQL L à N

Fonction C-SQL avec lien	Description
«LAST_DAY», à la page 147	Renvoie la date du dernier jour du mois contenant une date spécifiée
«LEAST», à la page 147	Renvoie la valeur la plus faible d'une liste d'expressions
«LOG», à la page 157	Renvoie le logarithme d'un nombre d'une base spécifique
«logNormDist», à la page 210	Renvoie la distribution log-normale cumulée d'une valeur
«LOWER», à la page 170	Convertit tous les caractères majuscules d'une chaîne en minuscules
«LPAD», à la page 170	Insère une ou plusieurs instances d'une chaîne au début d'une autre chaîne
«LTRIM», à la page 171	Supprime des caractères du début d'une chaîne
«MAX», à la page 179	Renvoie la valeur maximale dans un ensemble
«median», à la page 211	Renvoie la valeur médiane (centrale) d'un ensemble
«MIN», à la page 180	Renvoie la valeur minimale dans un ensemble
«MOD», à la page 158	Renvoie le modulo (reste) d'une division
«mode», à la page 212	Renvoie la valeur la plus fréquente d'un ensemble
«MOV_function», à la page 136	Limite les lignes utilisées pour le calcul d'une fonction d'ensembles aux lignes les plus récentes de la vue
«MOV_AVG», à la page 181	Renvoie la valeur moyenne (moyenne arithmétique) mobile d'un ensemble de valeurs numériques d'une fenêtre mobile
«MOV_COUNT», à la page 182	Renvoie le nombre de lignes d'un ensemble de fenêtres mobiles
«MOV_MAX», à la page 183	Retourne la valeur maximale d'un ensemble de fenêtre mobile

Tableau 52. Fonctions C-SQL L à N (suite)

Fonction C-SQL avec lien	Description
«MOV_MIN», à la page 184	Retourne la valeur minimale d'un ensemble de fenêtre mobile
«MOV_SUM», à la page 186	Renvoie la somme d'un ensemble de valeurs numériques d'une fenêtre mobile
«MOV_STD_DEVIATION», à la page 196	Renvoie l'écart type d'échantillonnage d'un ensemble de nombres de fenêtres mobiles
«MOV_VARIANCE», à la page 197	Renvoie le carré de l'écart type d'échantillonnage d'un ensemble de nombres de fenêtres mobiles
«NTILE», à la page 198	Détermine le niveau hiérarchique de chaque valeur d'un ensemble par rapport au reste de l'ensemble

Tableau 53. Fonctions C-SQL P à Z

Fonction C-SQL avec lien	Description
«POSITION», à la page 172	Renvoie la position d'un caractère ou d'une chaîne dans une chaîne
«POWER», à la page 159	Renvoie une valeur élevée à une puissance spécifique
«PREV», à la page 187	Renvoie une valeur de l'avant-dernière ligne d'un ensemble
«PRIOR_VALUE», à la page 148	Renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.
«RANK», à la page 200	Détermine le rang de chacune des valeurs d'un ensemble par rapport à l'intégralité de l'ensemble
«RATIO_TO_REPORT», à la page 201	Calcule le ratio d'une valeur par rapport à la somme des valeurs de l'ensemble
«ROUND», à la page 160	Renvoie un nombre arrondi à un nombre de décimales défini
«RPAD», à la page 173	Ajoute une ou plusieurs instances d'une chaîne à la fin d'une autre chaîne
«RTRIM», à la page 174	Supprime des caractères de la fin d'une chaîne
«SAFE_DIVIDE», à la page 161	Renvoie un quotient de deux valeurs, sauf si le quotient est égal à 0, auquel cas la fonction renvoie une autre valeur de quotient
«SIGN», à la page 162	Identifie le signe arithmétique d'un nombre
«SQRT», à la page 163	Renvoie la racine carrée d'un nombre
«STD_DEVIATION», à la page 202	Renvoie l'écart type d'échantillonnage d'un ensemble de nombres
«SUBSTRING», à la page 175	Renvoie la partie d'une chaîne identifiée par sa position et sa longueur
«SUM», à la page 188	Renvoie la somme d'un ensemble de valeurs numériques

Tableau 53. Fonctions C-SQL P à Z (suite)

Fonction C-SQL avec lien	Description
«SUM_OVER_GROUPS», à la page 203	Renvoie une somme évolutive des valeurs numériques classées en fonction de la colonne définie dans les arguments
«TIMESTAMP_DIFF», à la page 149	Renvoie l'intervalle de temps entre deux horodatages
«TO_CHAR», à la page 150	Convertit une date-heure en chaîne de caractères
«TO_DATE», à la page 151	Convertit une chaîne de caractères en une valeur date-heure
«TRUNC», à la page 163	Tronque une valeur après un nombre de décimales défini
«TUMBLE_AVG», à la page 189	Renvoie la valeur moyenne (moyenne arithmétique) d'un ensemble de fenêtres en défilement continu
«TUMBLE_COUNT», à la page 190	Renvoie le nombre de lignes d'un ensemble de fenêtres en défilement continu
«TUMBLE_MAX», à la page 191	Renvoie la valeur maximale d'un ensemble de fenêtres en défilement continu
«TUMBLE_MIN», à la page 193	Renvoie la valeur minimale d'un ensemble de fenêtres en défilement continu
«TUMBLE_SUM», à la page 194	Renvoie la somme d'un ensemble de fenêtres en défilement continu de valeurs numériques
«TUMBLE_STD_DEVIATION», à la page 204	Renvoie l'écart type d'échantillonnage d'un ensemble de nombres de fenêtres en défilement continu
«TUMBLE_VARIANCE», à la page 206	Renvoie le carré de l'exemple d'écart type d'un ensemble de valeurs d'une fenêtre en défilement continu
«UPPER», à la page 176	Convertit tous les caractères minuscules d'une chaîne en majuscules
«VARIANCE», à la page 207	Renvoie le carré de l'exemple d'écart type d'un ensemble de valeurs
«yield», à la page 213	Calcule le rendement donné d'un ensemble de résultats de test, et calcule le rendement

MOV_function

Les fonctions dont le format est `MOV_function` constituent des fonctions scalaires qui limitent les lignes utilisées dans le calcul des fonctions d'ensembles à l'ensemble des dernières lignes de la vue.

Le tableau suivant répertorie les sections qui fournissent des informations sur les fonctions des ensembles de fenêtres mobiles.

Tableau 54. Fonctions d'ensemble de fenêtre mobile

Argument	Description
«MOV_AVG», à la page 181	Renvoie la valeur moyenne (moyenne arithmétique) mobile d'un ensemble de valeurs numériques d'une fenêtre mobile.
«MOV_COUNT», à la page 182	Renvoie le nombre de lignes d'un ensemble d'une fenêtre mobile.
«MOV_MAX», à la page 183	Retourne la valeur maximale d'un ensemble de fenêtre mobile.
«MOV_MIN», à la page 184	Retourne la valeur minimale d'un ensemble de fenêtre mobile.
«MOV_SUM», à la page 186	Renvoie la somme d'un ensemble de valeurs numériques d'une fenêtre mobile.
«MOV_STD_DEVIATION», à la page 196	Renvoie l'écart type d'échantillonnage d'un ensemble de nombres de fenêtres mobiles.
«MOV_VARIANCE», à la page 197	Renvoie le carré de l'écart type d'échantillonnage d'un ensemble de nombres de fenêtres mobiles.

Les fonctions d'ensembles mobiles sont abrégées pour les fenêtres de requêtes simples. Pour obtenir une présentation complète, voir Chapitre 38, «Fenêtres de requêtes», à la page 391.

Syntaxe des fonctions MOV_function

Toutes les fonctions d'ensembles mobiles (Moving set) possèdent la même syntaxe :

```
mov_function(numeric, window, size [,timestampColumn])
```

Dans cet exemple, `mov_function` est une fonction d'ensembles (Set) existante. Le type de retour de la fonction mobile est identique à celui de la fonction nommée. La valeur numérique est généralement une colonne de la vue, mais elle peut contenir d'autres fonctions et opérateurs, même si elle ne peut pas faire référence à une fonction de classement (Rank).

Les arguments de taille et de fenêtre indiquent les lignes qui sont incluses dans l'ensemble. L'argument de fenêtre détermine si la taille représente le nombre de lignes de l'ensemble (EVENT) ou une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR). Par exemple, un ensemble des six derniers événements limite l'ensemble à six éléments par groupe si vous utilisez la clause GROUP BY. Pour en savoir davantage, voir «Clause GROUP BY», à la page 344. Certains événements génèrent plusieurs lignes.

La figure suivante présente une fonction d'ensembles mobiles (Moving set) qui renvoie la somme des six derniers prix, dans laquelle la fonction d'ensembles mobiles est définie sous la forme `MOV_SUM(Price, EVENT, 6)`. Etant donné que la fenêtre ne comportait que seuls cinq prix, la fonction renvoie la somme de ces cinq prix. Au fur et à mesure que la taille de la fenêtre augmente, la somme évolue, compte tenu que les prix les plus anciens ne sont pas inclus dans la somme, tel qu'indiqué dans les deux dernières fenêtres de l'illustration. L'expression `MOV_SUM(Price, EVENT, 6)` est identique à l'expression de fenêtre complète `SUM (Price) OVER (EVENT, '5' PRECEDING REFERENCE FRAME)`.

Figure 13. Fonction d'ensembles mobiles (Moving set) renvoyant la somme des six derniers prix

Lorsque vous faites appel à un intervalle chronologique (en lieu et place d'un intervalle d'événement), la taille de l'ensemble varie en fonction du moment où les événements ont été enregistrés dans la vue. Par exemple, si vous utilisez une série chronologique d'une heure, seules les lignes qui sont entrées dans la vue au cours de la dernière heure sont utilisées dans le calcul.

La figure suivante présente une fonction d'ensembles mobiles (Moving set) qui renvoie la somme des prix enregistrés au cours de la dernière heure, dans laquelle la fonction d'ensembles mobiles est définie sous la forme `MOV_SUM(PRICE, HOUR, 1)`. Etant donné que la fenêtre ne comportait initialement que cinq prix, la fonction renvoie la somme de ces cinq prix, car ils sont entrés dans la vue au cours de la dernière heure. Au fur et à mesure que la fenêtre s'agrandit, seuls les prix qui sont arrivés dans la vue au cours de la dernière heure sont inclus dans la somme. Comme illustré dans la figure suivante, une somme ultérieure inclut uniquement deux prix, car les prix précédents ne sont pas entrés dans la vue au cours de la dernière heure. L'expression `MOV_SUM(PRICE, HOUR, 1)` est identique à l'expression de fenêtre complète `SUM(Price) OVER (RANGE '1' HOUR PRECEDING REFERENCE OPERATOR)`.

Figure 14. Fonction d'ensembles mobiles (Moving set) renvoyant la somme des prix enregistrés au cours de la dernière heure

Avertissement : Il est possible qu'un événement arrive à temps pour être inclus dans une vue, mais qu'il soit annulé, car l'intervalle de temps est expiré au moment où il atteint la vue. Par exemple, si l'événement entre dans la table de flux de données quelques millisecondes avant l'intervalle planifié, il peut être inclus ou exclus en fonction de la durée nécessaire à son traitement et à sa propagation dans les vues de base.

L'argument facultatif timestampColumn indique au système d'utiliser la valeur d'une zone de la vue comme point de référence au départ de l'intervalle chronologique. Lorsque vous ignorez cette option, le système calcule la série chronologique en fonction de l'horloge système, par exemple la dernière heure à compter de maintenant. Lorsque vous attribuez un nom à la colonne d'horodatage à la place, le calcul est basé sur l'intervalle de temps à partir de la valeur qui est la plus récente dans la colonne de n'importe quelle ligne de la vue.

Dans la figure suivante, dans laquelle la fonction d'ensembles mobiles (Moving set) est définie sous la forme MOV_SUM(Price, HOUR, 1, Time), seuls deux prix sur six sont utilisés dans la somme, car les prix sélectionnés sont basés sur la valeur de la colonne Time et non sur leur ordre d'arrivée dans la vue. L'expression MOV_SUM(Price, HOUR, 1, Time) est équivalente à l'expression de fenêtre complète SUM (Price) OVER (ORDER BY Time RANGE '1' HOUR PRECEDING REFERENCE OPERATOR).

Figure 15. Fonction d'ensembles mobiles (Moving set) dans laquelle les prix sélectionnés sont basés sur la valeur de la colonne Time

Séries chronologiques

Les calculs de séries chronologiques utilisent le calendrier grégorien et reposent sur la seconde où l'événement a été enregistré dans la colonne vc_timestamp.

Par exemple, si la plage de temps est d'une journée et qu'un événement se produit juste avant minuit, il exclut la quasi-totalité des événements du jour qui précède. Toutefois, un événement survenant immédiatement après minuit inclut quasiment tous les événements de la veille. Voici quelques considérations supplémentaires :

Dans des fuseaux horaires appliquant une heure d'été, la durée des jours, mois et années est ajustée en conséquence. Même si un jour compte normalement 24 heures, il peut durer 23 ou 25 heures selon le moment.

Les calculs portant sur des mois reposent sur le jour du mois : une plage de temps d'un mois associée au 5 avril inclut toutes les dates qui suivent le 5 mars. Lorsque le jour du mois n'existe pas au début de la fenêtre, la fin du mois est utilisée. Par exemple, une plage de temps d'un mois associée au 31 mai démarre après le 30 avril.

De la même façon, les calculs portant sur des années reposent sur le jour de l'année et s'ajustent éventuellement en fonction des années bissextiles.

En cas d'exécution parallèle, vous pouvez choisir de ne pas traiter les événements dans l'ordre de leur horodatage. Dans ce cas, l'ordre des données au sein d'un groupe est arbitraire et ne produit que des valeurs d'ensembles mobiles (Moving set) approximatives non susceptibles de se reproduire pour les mêmes événements d'entrée au cours d'une évaluation ultérieure du même ensemble.

L'ensemble d'événements inclus dans une vue de fenêtre mobile est déterminé lorsqu'un nouvel événement entre dans la vue. Les événements filtrés avant leur arrivée dans la vue, par exemple du fait de leur exclusion par une clause WHERE n'ont pas de répercussions sur la vue et n'entraînent pas la mise à jour de cette dernière.

Avertissement concernant les vues

N'utilisez pas de fonction d'ensembles mobiles (Moving set) dans une vue dérivée pour effectuer un calcul sur le résultat d'une fonction d'ensembles mobiles dans une vue de base. En effet, la fonction dérivée renvoie toujours la valeur en cours dans la vue de base, indépendamment de la plage de temps de la fenêtre.

Si vous avez besoin d'une agrégation de ce type, placez les fonctions dans la même vue. Pour plus détails, voir l'exemple dans «AVG», à la page 177.

Interactions GROUP BY

L'utilisation d'une fonction d'ensembles mobiles (Moving set) dans une vue définie avec une clause GROUP BY remplit les groupes de la façon suivante.

Pour plus d'informations, voir «Clause GROUP BY», à la page 344.

Séries chronologiques

Les séries chronologiques s'appliquent à tous les événements de la totalité des groupes. Seuls les événements survenant au cours de cette plage de temps sont inclus dans les groupes. Les événements ne correspondant pas à la plage de temps définie sont exclus. Lorsque tous les flux de données sont supprimés d'un ensemble de groupes, ce groupe est vide. Si aucune autre colonne ne conserve ce groupe, ce dernier est supprimé de la vue. Examinez l'exemple ci-après recherchant les prix moyens dans des groupes par produit au cours de l'heure qui précède. Lorsqu'un produit ne comporte plus aucun événement pour l'heure qui vient de s'écouler, le groupe associé est supprimé.

```
SELECT MOV_AVG(Price,HOUR,1) AS Av_pr
  FROM ...
  GROUP BY Product
```

Toutefois, dans la variante ci-après, la présence de la fonction SUM() oblige la vue à conserver chacun des groupes de flux de données, mais il est possible que le prix moyen d'un groupe sur l'heure qui précède soit vide. L'envoi d'une requête à un groupe vide renvoie une valeur NULL.

```
SELECT MOV_AVG(Price,HOUR,1) AS Av_pr,
       SUM(Price) AS Total
  FROM ...
  GROUP BY Product
```

Plages de flux de données

Les plages de flux de données appliquent leur durée à chacun des groupes de la vue. Chaque groupe recherche un nombre de flux de données déterminé par la durée de la plage de temps. Les groupes ne sont jamais supprimés de la vue, et les flux de données ne sont supprimés des ensembles que lorsqu'ils sont sortis par un flux de données plus récent. Examinez la vue de l'exemple ci-après, dans laquelle MOV_AVG() recherche jusqu'à trois flux de données par groupe. Lorsque le quatrième flux de données présentant une valeur Ix égale à 100 est inséré, le premier flux de données est exclu du calcul de la moyenne mobile du groupe Ix=100. Toutefois, dans cet exemple, une fois que le groupe 200 est créé, l'ensemble reste constant avec ce même flux de données.

Figure 16. Une moyenne mobile recherche jusqu'à trois flux de données par groupe

Dans le cas des plages de flux de données englobant des flux de données constitués de plusieurs lignes dans la vue, la totalité du flux de données est considérée comme un seul élément dans l'ensemble basé sur l'horodatage (vc_timestamp) et sur l'ID du flux de données (vc_event_id).

Tous les noms columnNames directement désignés par une fonction de classement ou une fonction scalaire doivent apparaître dans l'ensemble de colonnes répertorié dans la liste GROUP BY.

Instruction CASE

Une instruction CASE évalue un ensemble de conditions et renvoie les résultats de l'une des expressions possibles.

L'instruction CASE est constituée des fonctions décrites dans les rubriques suivantes :

- «CASE»
- «ELSE», à la page 142
- «END», à la page 142
- «THEN», à la page 142
- «WHEN», à la page 142

CASE

Une expression CASE renvoie le résultat d'une expression correspondant à une condition vraie.

Chaque condition peut également renvoyer une valeur NULL. Si aucune condition vraie n'est trouvée, l'expression renvoie le résultat de la condition ELSE ou NULL si la condition ELSE est omise.

Pour obtenir des informations supplémentaires et des exemples, voir «Expression CASE», à la page 339.

ELSE

L'expression ELSE est renvoyée lorsque aucune opération de comparaison n'est évaluée comme étant vraie.

Si cet argument est omis et qu'aucune comparaison n'est vraie, l'instruction CASE renvoie la valeur NULL.

Pour obtenir des informations supplémentaires et des exemples, voir «Expression CASE», à la page 339.

END

La commande END permet de terminer une instruction CASE.

Pour obtenir des informations supplémentaires et des exemples, voir «Expression CASE», à la page 339.

THEN

Expression renvoyée lorsqu'une expression d'entrée correspond à une expression WHEN et qu'elle renvoie la valeur TRUE ou lorsqu'une expression booléenne renvoie la valeur TRUE.

Le résultat de l'expression peut être n'importe quelle expression C-SQL valide.

Pour obtenir des informations supplémentaires et des exemples, voir «Expression CASE», à la page 339.

WHEN

Expression simple à laquelle une expression d'entrée est comparée lors de l'utilisation du format CASE simple, ou expression booléenne évaluée lors de l'utilisation du format CASE de recherche.

Pour obtenir des informations supplémentaires et des exemples, voir «Expression CASE», à la page 339.

Conversion

Les fonctions de conversion permettent de convertir un type de données en un autre type de données.

CAST

Cette fonction scalaire (Scalar) convertit une valeur d'un type IBM Cognos Real-time Monitoring en un autre type Cognos Real-time Monitoring.

Syntaxe

```
CAST( value AS vcDataType )
```

Paramètres

- value

  Valeur à convertir.
- vcDataType

  L'un des types de données C-SQL (voir Chapitre 9, «Types de données», à la page 69) dans lequel la valeur doit être convertie.
  - INTEGER
  - DECIMAL
  - DOUBLE PRECISION
  - VARCHAR
  - TIMESTAMP
  - BOOLEAN
  - LONG

Type de valeur renvoyée

Même type de données que l'argument vcDataType.

Remarques

Les types sont convertis conformément au tableau «Ordre de priorité», à la page 70 fourni à la section «Conversion des types de données», à la page 69. CAST() renvoie une erreur si un type n'est pas convertissable tel qu'il est défini dans une expression. Par exemple, l'expression ci-après génère une erreur car C-SQL tente de convertir ‘4.5’ en valeur INTEGER, alors qu'une valeur décimale est un caractère incorrect pour les types de données INTEGER :

```
3 < CAST( '4.5' AS INTEGER )
```

En cas de conversion des valeurs décimales d'une colonne sous la forme d'une chaîne, des zéros sont ajoutés aux valeurs décimales de façon à correspondre à l'échelle, tout comme en cas de conversion d'une chaîne en une valeur décimale. Par exemple :

```
CAST( '1.1' AS DECIMAL(5,4) )  --> 1.1000
CAST( CAST( '1.1' AS DECIMAL(5,4) ) AS VARCHAR )  -->'1.1000'
```

Exemple

Convertir une chaîne de date en horodatage :

```
SELECT CAST('1997-10-22'
AS TIMESTAMP )
    FROM Cambrian;
```

Voir aussi

Pour plus de détails sur la conversion des différents types de données, voir «Conversion des types de données», à la page 69.

«TO_CHAR», à la page 150 convertit un horodatage en une chaîne de caractères d'un format défini.

«TO_DATE», à la page 151 convertit une chaîne de caractères en date.

Date et heure

Les fonctions de date et d'heure permettent de traiter les données associées à des dates, des heures et des horodatages.

CURRENT_TIMESTAMP

Cette fonction scalaire (Scalar) renvoie la date du jour et l'heure actuelle dans le fuseau horaire du serveur.

Syntaxe

```
CURRENT_TIMESTAMP()
```

Type de valeur renvoyée

Date-heure

Exemple

LAST_DAY(CURRENT_TIMESTAMP()) renvoie la date du dernier jour du mois en cours.

Voir aussi

«TO_CHAR», à la page 150 convertit un horodatage en chaîne de caractères.

«LAST_DAY», à la page 147 renvoie la date du dernier jour d'un mois.

«DATE_ADD» ajoute une durée à une valeur date-heure.

«DATE_DIFF», à la page 145 soustrait une durée d'une valeur date-heure.

«TIMESTAMP_DIFF», à la page 149 renvoie un intervalle de temps entre deux horodatages.

DATE_ADD

Cette fonction scalaire (Scalar) ajoute une durée à une valeur date-heure.

Syntaxe

```
DATE_ADD( timestamp, [ durationType, ] duration )
```

Paramètres

- timestamp
  Date-heure à ajuster.
- durationType
  Type de valeur de durée ; l'un des littéraux suivants :
  - SECOND
  - MINUTE
  - HOUR
  - DAY (valeur par défaut)
  - MONTH
  - YEAR

Type de valeur renvoyée

Date-heure

Remarques

Applique les règles d'ajout du calendrier grégorien.

Exemple

`DATE_ADD( CURRENT_TIMESTAMP(), 2)` renvoie la valeur date-heure obtenue après avoir ajouté deux jours à la date-heure en cours.

`DATE_ADD( timestamp, DAY, 14 )` renvoie la valeur obtenue après avoir ajouté 2 semaines à la date en cours.

Voir aussi

«DATE_DIFF» soustrait une durée d'une valeur date-heure.

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

«TIMESTAMP_DIFF», à la page 149 renvoie un intervalle de temps entre deux horodatages.

DATE_DIFF

Cette fonction scalaire (Scalar) soustrait une durée d'une valeur date-heure.

Syntaxe

```
DATE_DIFF( timestamp, [ durationType, ] duration )
```

Paramètres

- timestamp
  Valeur date-heure de laquelle doit être soustraite la durée.
- durationType
  Type de valeur de durée ; l'un des littéraux suivants :
  - SECOND
  - MINUTE
  - HOUR
  - DAY (valeur par défaut)
  - MONTH
  - YEAR

Type de valeur renvoyée

Date-heure

Remarques

Applique les règles de soustraction du calendrier grégorien.

Les durées englobant des jours d'années bissextiles et des secondes intercalaires ignorent généralement la valeur d'ajustement. Par exemple, si la fonction retranche une année de la date du 3 mars 1976, elle renvoie la valeur 3 mars 1975 sans tenir compte du 29 février 1976. Toutefois, si la fonction retranche une année de la date du 29 février, elle renvoie la date du 28 février.

Exemples

DATE_DIFF(CURRENT_TIMESTAMP(), 2) renvoie la valeur date-heure obtenue après avoir retranché deux jours de la date-heure en cours.

Voir aussi

«DATE_ADD», à la page 144 ajoute une durée à une valeur date-heure.

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

«TIMESTAMP_DIFF», à la page 149 renvoie un intervalle de temps entre deux horodatages.

GREATEST

Cette fonction scalaire (Scalar) renvoie le plus grand des résultats d'expression de la liste.

Syntaxe

```
GREATEST( value, value [, value ... ] )
```

Paramètre

- value

  Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

Sélectionner la chaîne présentant la valeur maximale :

```
SELECT Greatest(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Students;
Greatest
--------
SKYLER
```

Voir aussi

«LEAST» détermine la valeur minimale d'une liste.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

La section «Types de fonction», à la page 125 présente les ensembles mobiles.

LAST_DAY

Cette fonction scalaire (Scalar) renvoie la date du dernier jour du mois contenant une date précise.

Syntaxe

```
LAST_DAY( dateTime )
```

Paramètre

- dateTime

  Date valide (TIMESTAMP).

Type de valeur renvoyée

Date-heure

Exemple

LAST_DAY(CURRENT_TIMESTAMP()) renvoie la date du dernier jour du mois en cours.

Voir aussi

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

LEAST

Cette fonction scalaire (Scalar) renvoie la valeur la plus faible d'une liste d'expressions.

Syntaxe

```
LEAST( value, value [, value ... ] )
```

Paramètre

- value

  Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

```
SELECT Least(
        'SCHOLAR',
        'SKYLER',
```

```
        'SHUELLER')
    FROM Students;
Least
------
SCHOLAR
```

Voir aussi

«GREATEST», à la page 146 détermine la valeur maximale d'une liste.

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble.

La section «Types de fonction», à la page 125 présente les ensembles mobiles.

PRIOR_VALUE

Cette fonction scalaire (Scalar) renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.

Syntaxe

```
PRIOR_VALUE( columnName )
```

Paramètre

- columnName
  Colonne ou alias d'un type de données quelconque à évaluer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie une valeur NULL si aucune valeur précédente n'existe, c'est-à-dire lors de la première exécution de la fonction sur la valeur columnName spécifiée.

Utilisez la fonction PRIOR_VALUE() lorsque les données (flux de données) entrent dans le système de façon groupée et triée.

Cette fonction n'est pas autorisée dans la clause WHERE (voir «Clause WHERE», à la page 343) d'une définition de vue.

Exemple

Examinez la requête ci-après identifiant, sous la forme d'un pourcentage d'une heure, la durée requise pour l'exécution d'une tâche, en fonction du nombre de minutes écoulées depuis l'exécution de la tâche précédente :

```
SELECT Task, CAST(
      (TIMESTAMP_DIFF(PRIOR_VALUE(Completed),Completed,
MINUTE ) /60
                  , DECIMAL(5,4)
                 ) AS Hours
    FROM Tasks_Completed
Task            Hours
---------------  ------
```

```
Startup
Initialize       0.0887
Begin job        0.1012
Finish job       4.3243
Clean up         0.2500
Shut down        0.1285
Have milk shake  0.6667
```

Voir aussi

«PREV», à la page 187 renvoie une valeur de l'avant-dernière ligne d'un ensemble.

TIMESTAMP_DIFF

Cette fonction scalaire (Scalar) renvoie, sous la forme d'une valeur absolue, l'intervalle de temps entre deux horodatages.

Syntaxe

```
TIMESTAMP_DIFF( startTime, endTime, unit )
```

Paramètres

- startTime
  Date-heure de début.
- endTime
  Date-heure de fin.
- unit
  Type d'intervalle de temps à renvoyer ; l'un des littéraux suivants :
  - SECOND
  - MINUTE
  - HOUR
  - DAY MONTH (30 jours) YEAR (12 mois ou 360 jours)

Type de valeur renvoyée

INTEGER

Remarques

Cette fonction utilise la différence de temps absolue arrondie au nombre entier supérieur le plus proche ; elle n'utilise pas l'arithmétique du calendrier grégorien.

La fonction arrondit le résultat au nombre entier le plus proche. Par exemple, la différence entre 10:00 et 10:29 en unités de type HOUR est égale à zéro (0), mais la différence entre 10:00 et 10:30 renvoie la valeur un (1).

Renvoie le résultat NULL si l'un ou l'autre horodatage a la valeur NULL.

Exemple

Renvoyer le nombre de jours entre le moment présent et la fin du mois.

```
TIMESTAMP_DIFF( CURRENT_TIMESTAMP(),
               LAST_DAY( CURRENT_TIMESTAMP() ), DAY)
```

Renvoyer la valeur True lorsqu'un dossier-problème est ouvert pendant plus de 30 minutes :

```
TIMESTAMP_DIFF( ticket_opened, CURRENT_TIMESTAMP(), MINUTE)
> 30
```

Renvoyer le nombre de jours entre deux dates sous la forme d'un nombre positif, quelle que soit la date la plus ancienne :

```
ABS( TIMESTAMP_DIFF( father_birthdate, mother_birthdate,
DAY) )
```

Voir aussi

«DATE_ADD», à la page 144 ajoute une durée à une valeur date-heure.

«DATE_DIFF», à la page 145 soustrait une durée d'une valeur date-heure.

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

TO_CHAR

Cette fonction scalaire (Scalar) convertit une date-heure en chaîne de caractères.

Syntaxe

```
TO_CHAR( date, [ format ] )
```

Paramètres

- date

  Valeur date-heure (voir «Date-heure», à la page 76) à convertir.
- format

  Chaîne de modèle de date identique à celle utilisée par la classe Java SimpleDateFormat et décrite à la section «Format date-heure», à la page 81. N'indiquez pas cette option si vous souhaitez convertir la valeur en utilisant le format par défaut, à savoir "aaaa-MM-jj HH:mm:ss.SSSSSSSSS".

Type de valeur renvoyée

VARCHAR

Remarques

Pour plus d'informations sur la conversion, voir «Conversion des valeurs date-heure en chaînes», à la page 76.

Exemples

TO_CHAR(CURRENT_TIMESTAMP(), 'd MMMM yy') renvoie '5 March 03'.

Voir aussi

«CAST», à la page 142 convertit un type de données en un autre type.

«TO_DATE», à la page 151 convertit une chaîne de caractères en date.

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

TO_DATE

Cette fonction scalaire (Scalar) convertit une chaîne de caractères en valeur date-heure.

Syntaxe

```
TO_DATE( string, [ format ] )
```

Paramètres

- string

  Chaîne de date (VARCHAR) ou littéral à convertir.
- format

  Chaîne de modèle de date identique à celle utilisée par la classe Java SimpleDateFormat et décrite à la section «Format date-heure», à la page 81. N'indiquez pas cette option si vous souhaitez convertir la valeur en utilisant le format par défaut, à savoir "yyyy-MM-dd HH:mm:ss.SSSSSSSSS".

Type de valeur renvoyée

Date-heure

Remarques

Si vous n'indiquez aucune valeur d'heure dans le modèle, la fonction insère des zéros (0) dans la partie correspondante des données TIMESTAMP, configurant ainsi l'heure sur minuit.

Exemples

TO_DATE('2003-02-18') produit le même résultat que TO_DATE('2003-02-18', "yyyy-MM-dd").

TO_DATE('2003-02-18 12:00:00', 'yyyy-MM-dd HH:mm:ss') configure l'heure sur midi.

Pour supprimer la partie heure d'une valeur TIMESTAMP, convertissez-la en caractères, puis de nouveau en date :

```
TO_DATE( TO_CHAR( a_timestamp, 'yyyy-MM-dd' ) )
```

Voir aussi

«CAST», à la page 142 convertit un type de données en un autre type.

«TO_CHAR», à la page 150 convertit une date en chaîne de caractères.

«CURRENT_TIMESTAMP», à la page 144 renvoie la date du jour et l'heure en cours.

Nombres

Les fonctions de numération permettent d'effectuer des opérations sur des données numériques.

ABS

Cette fonction scalaire (Scalar) renvoie la valeur absolue d'un nombre.

Syntaxe

```
ABS( numeric )
```

Paramètre

- numeric

  Expression évaluée sous la forme d'une valeur numérique.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric.

Exemple

Renvoyer la différence d'âge de deux personnes, quelle que soit la plus âgée des deux.

```
SELECT ABS( father_age - mother_age ) AS "Difference
of parents ages"
    FROM Family
```

Voir aussi

«SIGN», à la page 162 renvoie le signe arithmétique d'un nombre.

CEIL

Cette fonction scalaire (Scalar) renvoie le plus petit nombre entier, arrondi à partir de zéro, supérieur ou égal à un nombre.

Syntaxe

```
CEIL( numeric )
```

Paramètre

- numeric

  Nombre à arrondir.

Type de valeur renvoyée

Même type de données que l'argument numeric.

Exemple

CEIL(1234.56) renvoie 1235.00.

CEIL(-2.75) renvoie -2.00.

Voir aussi

«FLOOR», à la page 155 renvoie la valeur maximale inférieure ou égale à un nombre.

DISPLAY_MONEY

Cette fonction scalaire (Scalar) formate un nombre en tant que valeur de devise.

Syntaxe

```
DISPLAY_MONEY( number [, languageCode, countryCode ])
```

Paramètres

- number

  Nombre à formater.
- languageCode

  Code de langue ISO 639 à deux lettres. Permet de déterminer le symbole de devise à afficher.
- countryCode

  Code de pays ISO 3166 à deux lettres. Indique le séparateur de milliers, le séparateur décimal, ainsi que le nombre de chiffres décimaux à afficher en fonction des conventions en usage dans le pays.

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie un nombre présenté sous la forme d'une chaîne de devise.

L'omission des paramètres languageCode et countryCode entraîne l'utilisation du symbole et du format adaptés au pays que votre ordinateur est configuré pour utiliser par défaut.

Certains symboles de devise nécessitent que le navigateur soit configuré sur la page de code appropriée pour la langue.

Codes courants

Le tableau ci-après répertorie les codes de langue ISO 639 à deux lettres les plus courants :

Tableau 55. Codes de pays ou de région courants

Langue	Code	Langue	Code	Langue	Code
Afrikaans	af	Fidjien	fj	Néerlandais	nl
Arabe	ar	Féroïen	fo	Norvégien	non
Catalan	ca	Français	fr	Panjabi	pa
Corse	co	Hébreu	he	Polonais	pl
Tchèque	cs	Hindi	hi	Portugais	pt
Danois	da	Croate	h	Russe	ru
Allemand	de	Italien	it	Serbo-croate	sh
Grec	el	Inuktitut	iu	Suédois	sv
Anglais	en	Japonais	ja	Turc	tr
Espagnol	es	Coréen	ko	Urdu	ur
Persan	fa	Mongol	mn	Yoruba	yo

Tableau 55. Codes de pays ou de région courants (suite)

Langue	Code	Langue	Code	Langue	Code
Finnois	fi	Népalais	ne	Chinois	zh

Le tableau ci-après répertorie les codes de pays ISO 3166 à deux lettres les plus courants.

Tableau 56. Codes de pays ou de région ISO 3166 à deux lettres courants

Pays ou région	Code	Pays ou région	Code
AUSTRALIE	AU	NEPAL	NP
AUTRICHE	AT	PAYS-BAS	NL
BRESIL	BR	NOUVELLE-ZELANDE	NZ
CANADA	CA	NORVEGE	NO
CHINE	CN	OMAN	OM
CROATIE (nom local : Hrvatska)	HR	PAKISTAN	PK
DANEMARK	DK	PITCAIRN	PN
FIDJI	FJ	POLOGNE	PL
FINLANDE	FI	PORTUGAL	PT
FRANCE	FR	ARABIE SAOUDITE	SA
ALLEMAGNE	DE	AFRIQUE DU SUD	ZA
GRECE	GR	ESPAGNE	ES
INDE	IN	SUISSE	CH
IRAN (République islamique)	IR	THAILANDE	TH
ISRAEL	IL	TURQUIE	TR
ITALIE	IT	EMIRATS ARABES UNIS	AE
JAPON	JP	ROYAUME-UNI	GB
COREE (République populaire démocratique)	KP	ETATS-UNIS	US
MEXIQUE	MX	--	--

Exemples

Le tableau ci-après présente quelques exemples de mise en forme du nombre 12345.678 :

Tableau 57. Exemples de mise en forme du nombre 12345.678

Langue/Pays	Formule	Résultat
Anglais/Etats-Unis	DISPLAY_MONEY(12345.678,'en','us')	$12,345.68
Suédois/Suède	DISPLAY_MONEY(12345.678,'sv','se')	12 345,68 kr
Allemand/ Allemagne	DISPLAY_MONEY(12345.678,'de','de')	12.345,68 dm

EXP

Cette fonction scalaire (Scalar) renvoie e élevé à une puissance spécifique.

Syntaxe

```
EXP( power )
```

Paramètre

- power
  Puissance (DOUBLE) à laquelle *e* doit être élevé.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie *e* élevé à la puissance n^e puissance, où *e* = 2.71828183...

Exemple

EXP(4) élève *e* à la 4e puissance et renvoie 54.59815.

Voir aussi

«LOG», à la page 157 renvoie le logarithme d'un nombre d'une base spécifique.

«POWER», à la page 159 élève une valeur à une puissance spécifique.

FLOOR

Cette fonction scalaire (Scalar) renvoie le plus grand nombre entier inférieur ou égal à une expression.

Syntaxe

```
FLOOR( numeric )
```

Paramètre

- numeric
  Nombre à arrondir au chiffre inférieur.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric.

Exemple

FLOOR('1234.56') renvoie le nombre entier 1234.00, après avoir commencé par convertir implicitement le littéral de chaîne en valeur de type DECIMAL.

FLOOR(-2.75) renvoie -3.00.

Cette fonction produit le même résultat que la fonction Microsoft Excel INT().

Voir aussi

«CEIL», à la page 152 renvoie le plus petit nombre entier arrondi au chiffre supérieur.

GREATEST

Cette fonction scalaire (Scalar) renvoie le plus grand des résultats d'expression de la liste.

Syntaxe

```
GREATEST( value, value [, value ... ] )
```

Paramètre

- value

  Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

Sélectionner la chaîne présentant la valeur maximale :

```
SELECT Greatest(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Students;
Greatest
--------
SKYLER
```

Voir aussi

«LEAST», à la page 147 détermine la valeur minimale d'une liste.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

LEAST

Cette fonction scalaire (Scalar) renvoie la valeur la plus faible d'une liste d'expressions.

Syntaxe

```
LEAST( value, value [, value ... ] )
```

Paramètre

- value

Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

```
SELECT Least(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Students;
Least
------
SCHOLAR
```

Voir aussi

«GREATEST», à la page 146 détermine la valeur maximale d'une liste.

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble.

La section «Types de fonction», à la page 125 présente les ensembles mobiles.

LOG

Cette fonction scalaire (Scalar) renvoie le logarithme d'un nombre d'une base spécifique.

Syntaxe

```
LOG( numeric [, base ] )
```

Paramètres

- numeric

  Nombre (DOUBLE) dont le logarithme doit être calculé ; cette valeur doit être supérieure à 1.
- base

  Base (DOUBLE) du logarithme ; cette valeur doit être supérieure à zéro (0). N'utilisez pas cette option si vous souhaitez récupérer le logarithme naturel d'une valeur numérique.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Cette fonction peut être exprimée de façon mathématique sous la forme $\log_{numeric}{}^{base}$.

Exemple

LOG(8,64) renvoie 2.0.

LOG(2) renvoie 0.301029...

Voir aussi

«EXP», à la page 155 renvoie e élevé à une puissance spécifique.

«POWER», à la page 159 élève une valeur à une puissance spécifique.

MOD

Cette fonction scalaire (Scalar) renvoie le modulo (reste) d'une division.

Syntaxe

```
MOD( dividend, divisor )
```

Paramètres

- dividend
  Dividende.
- divisor
  Diviseur.

Type de valeur renvoyée

INTEGER

Remarques

Lorsque le diviseur est égal à zéro (0), cette fonction renvoie le dividende.

Cette fonction ne produit pas le même résultat que la fonction de modulo mathématique classique lorsque le dividende est négatif. Le modulo classique peut être exprimé à l'aide de la formule suivante :

```
dividend - divisor * FLOOR(dividend/divisor)
```

La présente fonction utilise la formule modifiée suivante :

```
SIGN(dividend) *
     (ABS(dividend) - ABS(divisor * FLOOR(ABS(dividend/divisor))))
```

Le tableau ci-après indique les différences entre la fonction MOD et la formule de modulo classique :

Tableau 58. Différences entre la fonction MOD et la formule de modulo classique

Dividende	Diviseur	MOD (dividend,divisor)	Modulo classique
11	4	3	3
11	-4	3	-1
-11	4	-3	1
-11	-4	-3	-3

La plupart des systèmes de gestion de base de données utilisent la même formule qu'IBM Cognos Real-time Monitoring, tandis que les tableurs tels que Microsoft Excel utilisent la fonction modulo classique.

Exemple

Renvoyer le reste du dividende divisé par le diviseur :

```
SELECT MOD(11,4) "Modulus" FROM Census;
Modulus
-------
3
```

POWER

Cette fonction scalaire (Scalar) renvoie une valeur élevée à une puissance spécifique.

Syntaxe

```
POWER( numeric, power )
```

Paramètres

- numeric

  Nombre à élever à une puissance.
- power

  Puissance à laquelle la valeur numérique doit être élevée. Il doit s'agir d'un nombre entier lorsque la valeur numérique est négative.

Type de valeur renvoyée

Même type de données que l'argument numeric.

Remarques

Cette fonction peut être exprimée de façon mathématique sous la forme nombre puissance.

Exemple

POWER(3,5) renvoie 243.

Voir aussi

«EXP», à la page 155 renvoie e élevé à une puissance spécifique.

«LOG», à la page 157 renvoie le logarithme d'un nombre d'une base spécifique.

PRIOR_VALUE

Cette fonction scalaire (Scalar) renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.

Syntaxe

```
PRIOR_VALUE( columnName )
```

Paramètre

- columnName

  Colonne ou alias d'un type de données quelconque à évaluer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie une valeur NULL si aucune valeur précédente n'existe, c'est-à-dire lors de la première exécution de la fonction sur la valeur columnName spécifiée.

Utilisez la fonction PRIOR_VALUE() lorsque les données (flux de données) entrent dans le système de façon groupée et triée.

Cette fonction n'est pas autorisée dans la «Clause WHERE», à la page 343 d'une définition de vue.

Exemple

Examinez la requête ci-après identifiant, sous la forme d'un pourcentage d'une heure, la durée requise pour l'exécution d'une tâche, en fonction du nombre de minutes écoulées depuis l'exécution de la tâche précédente :

```
SELECT Task, CAST(
      (TIMESTAMP_DIFF(PRIOR_VALUE(Completed),Completed,
MINUTE ) /60
                  , DECIMAL(5,4)
                 ) AS Hours
    FROM Tasks_Completed
Task             Hours
---------------  ------
Startup
Initialize       0.0887
Begin job        0.1012
Finish job       4.3243
Clean up         0.2500
Shut down        0.1285
Have milk shake  0.6667
```

Voir aussi

«PREV», à la page 187 renvoie une valeur de l'avant-dernière ligne d'un ensemble.

ROUND

Cette fonction scalaire (Scalar) renvoie un nombre arrondi à un nombre de décimales défini.

Syntaxe

```
ROUND( number, [ places ] )
```

Paramètres

- number

  Expression numérique à arrondir.
- places

  Nombre de décimales auquel le nombre doit être arrondi. Lorsque le nombre de décimales est un nombre entier négatif, la fonction renvoie un nombre entier rond. La valeur par défaut de ce paramètre est zéro (0), ce qui a pour effet de supprimer toutes les décimales.

Type de valeur renvoyée

Numérique, même type de données que l'argument number.

Exemples

ROUND(1294.5078) renvoie 1294.

ROUND(1294.5078, 0) renvoie 1294.

ROUND(1294.5078, 1) renvoie 1294.5.

ROUND(1294.5078, 2) renvoie 1294.51.

ROUND(1294.5078, -2) renvoie 1300.

Voir aussi

«TRUNC», à la page 163 supprime des chiffres d'un nombre.

SAFE_DIVIDE

Cette fonction scalaire (Scalar) renvoie un quotient déterminé par un dividende et un diviseur, ou un quotient de remplacement en cas de division par zéro.

Si le diviseur est différent de zéro, cette fonction renvoie le quotient de la division du dividende par le diviseur. Si le diviseur est égal à zéro ou qu'une autre erreur d'exécution survient pendant le calcul du quotient, par exemple dans le cas d'une valeur DECIMAL hors limites, la fonction renvoie le quotient de remplacement.

Il est conseillé d'utiliser des valeurs décimales à la fois pour le dividende et pour le diviseur.

Syntaxe

```
SAFE_DIVIDE(numberDividend,numberDivisor,numberAlternateQuotient)
```

Paramètres

- numberDividend

  Valeur numérique à diviser.

  Le type de données est déterminé par la façon dont la valeur est entrée. Si la valeur est entrée sans décimale (par exemple, 19), cette valeur est convertie en nombre entier. Si la valeur est entrée avec décimale (par exemple, 19.00), cette valeur est convertie en décimale.
- numberDivisor

  ’aleur numérique par laquelle le dividende doit être divisé.

Cette valeur est automatiquement convertie dans le même type de données que la valeur numberDividend.

- numberAlternateQuotient

  Valeur numérique à renvoyer si le quotient obtenu est égal à 0 ou si une autre erreur d'exécution se produit lors du calcul du quotient.

Type de valeur renvoyée

Numérique, même type de données que l'argument number.

Exemples

`SAFE_DIVIDE (100,20,4321)` renvoie 5.

`SAFE_DIVIDE (100.19.5,4321)` renvoie 5. valeur 19.5 est convertie en nombre entier, c'est-à-dire 19.

`SAFE_DIVIDE (100.00,19.5,4321)` renvoie 5.12. Le quotient est renvoyé avec le même nombre de décimales que la valeur numberDividend, deux décimales dans le cas présent.

`SAFE_DIVIDE (SAFE_DIVIDE(100.000,19.5,4321)` renvoie 5.128. Le quotient est renvoyé avec le même nombre de décimales que la valeur numberDividend, trois décimales dans le cas présent.

`SAFE_DIVIDE (100.0,4321)` renvoie 4321. Le quotient de la division de numberDividend par numberDivisor étant égal à 0, la fonction renvoie la valeur numberAlternateQuotient.

SIGN

Cette fonction scalaire (Scalar) identifie le signe arithmétique d'un nombre.

Syntaxe

```
SIGN( number )
```

Paramètre

- number

  Valeur numérique à évaluer.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie la valeur 1 si le nombre est positif, la valeur 0 si le nombre est égal à zéro et la valeur -1 si le nombre est négatif.

Notez que les expressions ci-après renvoient des résultats identiques :

```
(number * SIGN(number) )
ABS(number)
```

Voir aussi

«ABS», à la page 152 renvoie la valeur absolue d'un nombre.

SQRT

Cette fonction scalaire (Scalar) renvoie la racine carrée d'un nombre.

Syntaxe

```
SQRT( number )
```

Paramètre

- number

  Valeur numérique (DOUBLE) à évaluer. Cette valeur doit être supérieure à zéro (0).

Type de valeur renvoyée

DOUBLE PRECISION

Exemple

SQRT(42) renvoie 6.480...

TRUNC

Cette fonction scalaire (Scalar) tronque une valeur après un nombre de décimales défini.

Lorsqu'il s'agit d'un horodatage, cette fonction renvoie zéro dans les zones en dessous de la précision donnée.

Syntaxe

```
TRUNC( decimalNumber [, places ] )
TRUNC( timestamp [, 'D' | 'h' | 'm' | 's' ] )
```

Paramètres

- decimalNumber

  Nombre à tronquer.
- timestamp

  Horodatage à tronquer.
- places

  Nombre de décimales en fonction duquel le nombre doit être tronqué. Si vous omettez cette valeur, la fonction tronque toutes les décimales et renvoie un nombre entier. Si cette valeur est négative, la fonction convertit les chiffres en zéros.
- 'D'

  Renvoie uniquement les informations relative au jour dans l'horodatage. Les heures, les minutes et les secondes sont renvoyées sous la forme de zéros.
- 'h'

  Renvoie uniquement les informations relatives à l'heure dans l'horodatage. Les minutes et les secondes sont renvoyées sous la forme de zéros.
- 'm'

Renvoie uniquement les informations relatives au jour, à l'heure et aux minutes dans l'horodatage. Les secondes sont renvoyées sous la forme de zéros.

- 's'

  Renvoie uniquement les informations relatives au jour, à l'heure et aux secondes dans l'horodatage, mais pas aux millisecondes.

Type de valeur renvoyée

Numérique, même type de données que l'argument decimalNumber.

Exemples

TRUNC(1234.567) renvoie 1234.

TRUNC(1234.567, 1) renvoie 12345.6.

TRUNC(1234.567, -2) renvoie 1200.

TRUNC(2007-06-27 11:01:414.213, 'D') renvoie 2007-06-27 0:0:0.0

TRUNC(2007-06-27 11:01:414.213, 'D 'h') renvoie 2007-06-27 11:0:0.0

TRUNC(2007-06-27 11:01:414.213, 'D 'h' 'm') renvoie 2007-06-27 11:01:0.0

TRUNC(2007-06-27 11:01:414.213, 'D 'h' 'm' 's') renvoie 2007-06-27 11:01:213.0

Voir aussi

«ROUND», à la page 160 arrondit un nombre à un nombre de décimales défini.

IS_ACKED

Cette fonction scalaire (Scalar) renvoie une valeur vraie (True) lorsque l'alerte spécifiée a été reçue.

Syntaxe

```
IS_ACKED('alertName')
```

Paramètre

- alertName

  Nom qualifié complet d'une alerte (voir Chapitre 5, «Alertes», à la page 43). Ce nom doit inclure le nom de l'activité métier et le nom du scénario associés, comme suit : 'NomActivité.NomScénario.NomAlerte'.

Type de valeur renvoyée

Booléen

Remarques

Renvoie la valeur vraie (True) si l'alerte existe et qu'un accusé de réception a été reçu. Dans le cas contraire, renvoie la valeur fausse (False).

Cette fonction renvoyant la valeur False lorsque l'alerte n'existe pas, aucun contrôle n'est effectué pour vérifier la validité du nom d'alerte spécifié dans le système. Par conséquent, la fonction renvoie toujours la valeur False si le nom de l'alerte n'est pas correctement orthographié.

Utilisez cette fonction dans une condition de règle pour tester l'état d'une alerte et pour générer une nouvelle alerte lorsque l'alerte testée ne reçoit pas d'accusé de réception sur une période donnée.

Si vous utilisez cette fonction dans une définition de vue, cette dernière ne peut pas comporter de fonction d'ensembles.

IS_RAISED

Cette fonction scalaire (Scalar) renvoie la valeur TRUE lorsque l'alerte spécifiée présente l'état lancé.

Syntaxe

```
IS_RAISED('alertName')
```

Paramètre

- alertName

  Nom qualifié complet d'une alerte (voir Chapitre 5, «Alertes», à la page 43). Ce nom doit inclure le nom de l'activité métier et le nom du scénario associés, comme suit : 'NomActivité.NomScénario.NomAlerte'.

Type de valeur renvoyée

Booléen

Remarques

Renvoie la valeur True si l'alerte existe et présente l'état lancé. Renvoie la valeur False si l'alerte est à l'état diminué ou inexistante.

Cette fonction renvoyant la valeur False lorsque l'alerte n'existe pas, aucun contrôle n'est effectué pour vérifier la validité du nom d'alerte spécifié dans le système. Par conséquent, la fonction renvoie toujours la valeur False si le nom de l'alerte n'est pas correctement orthographié.

Utilisez cette fonction dans une condition de règle pour tester l'état d'une alerte et pour générer une nouvelle alerte lorsque l'alerte testée conserve l'état élevé sur une période donnée.

Si vous utilisez cette fonction dans une définition de vue, cette dernière ne peut pas comporter de fonction d'ensembles.

Pour en savoir davantage, voir «Règles contrôlant les alertes», à la page 325.

PRIOR_VALUE

Cette fonction scalaire (Scalar) renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.

Syntaxe

```
PRIOR_VALUE( columnName )
```

Paramètre

- columnName
  Colonne ou alias d'un type de données quelconque à évaluer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie une valeur NULL si aucune valeur précédente n'existe, c'est-à-dire lors de la première exécution de la fonction sur la valeur columnName spécifiée.

Utilisez la fonction PRIOR_VALUE() lorsque les données (flux de données) entrent dans le système de façon groupée et triée.

Cette fonction n'est pas autorisée dans la «Clause WHERE», à la page 343 d'une définition de vue.

Exemple

Examinez la requête ci-après identifiant, sous la forme d'un pourcentage d'une heure, la durée requise pour l'exécution d'une tâche, en fonction du nombre de minutes écoulées depuis l'exécution de la tâche précédente :

```
SELECT Task, CAST(
      (TIMESTAMP_DIFF(PRIOR_VALUE(Completed),Completed,
MINUTE ) /60
                   , DECIMAL(5,4)
                  ) AS Hours
    FROM Tasks_Completed
Task             Hours
---------------  ------
Startup
Initialize       0.0887
Begin job        0.1012
Finish job       4.3243
Clean up         0.2500
Shut down        0.1285
Have milk shake  0.6667
```

Voir aussi

«PREV», à la page 187 renvoie une valeur de l'avant-dernière ligne d'un ensemble.

Système

La fonction inhérente à cette catégorie permet de renvoyer un nom de connexion.

CURRENT_USER

Cette fonction scalaire (Scalar) renvoie le nom de connexion de l'utilisateur actuel.

Syntaxe

```
CURRENT_USER()
```

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie le nom de connexion de l'utilisateur tel qu'il est défini dans IBM Cognos Real-time Monitoring, en respectant la distinction majuscules-minuscules, et tel qu'il apparaît dans Cognos Real-time Monitoring Workbench. En cas d'utilisation de ce nom dans une comparaison, la distinction majuscules-minuscules doit donc être scrupuleusement respectée.

Cette fonction est essentiellement destinée à être utilisée dans les filtres d'accès. Pour découvrir des exemples et des utilisations possibles, voir Chapitre 2, «Filtres d'accès», à la page 3, et plus précisément «Tables de correspondance utilisateur dans les filtres d'accès», à la page 4.

Autre

Ces fonctions peuvent être utilisées avec les alertes.

Texte

Les fonctions de texte permettent d'effectuer des opérations sur les chaînes.

CHAR_LENGTH

Cette fonction scalaire (Scalar) renvoie la longueur d'une chaîne.

Syntaxe

```
CHARACTER_LENGTH( string )
```

Paramètre

- string
  Chaîne ou résultat d'expression VARCHAR dont la longueur doit être évaluée.

Type de valeur renvoyée

INTEGER

Remarques

Vous pouvez également orthographier cette fonction sous la forme :

```
CHAR_LENGTH( string )
```

Renvoie un nombre entier indiquant la longueur de la chaîne. Renvoie NULL si la chaîne a la valeur NULL.

La longueur d'une chaîne est déterminée par le nombre de caractères affichables de la chaîne, et non nécessairement par la capacité de stockage requise pour la chaîne. Par exemple, un caractère Unicode requiert 16 bits pour son stockage, ce que certains systèmes peuvent considérer comme une longueur de stockage de deux

caractères, mais la longueur réelle de ce caractère est égale à 1.

CONCAT

Cette fonction scalaire (Scalar) renvoie une chaîne correspondant à la concaténation de deux caractères ou chaînes.

Syntaxe

```
CONCAT( string1, string2 )
```

Paramètre

- string

  Valeur de chaîne de caractères ou résultat d'expression VARCHAR.

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie la valeur string2 ajoutée à la fin de string1. Renvoie NULL si l'une ou l'autre chaîne a la valeur NULL.

L'opérateur `||` («Opérateurs de chaîne», à la page 291) Renvoie NULL si l'une ou l'autre chaîne a la valeur.

Exemples

CONCAT('a', 'b') renvoie 'ab'.

'a'||'b' renvoie 'ab'.

Voir aussi

«concatList», à la page 208 renvoie une chaîne correspondant à la concaténation d'une liste de caractères ou de chaînes.

«concatSet», à la page 209 renvoie un ensemble de chaînes trié dans l'ordre alphabétique.

La section «Opérateurs de chaîne», à la page 291 décrit l'opérateur `||`.

GREATEST

Cette fonction scalaire (Scalar) renvoie le plus grand des résultats d'expression de la liste.

Syntaxe

```
GREATEST( value, value [, value ... ] )
```

Paramètre

- value

  Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

Sélectionner la chaîne présentant la valeur maximale :

```
SELECT Greatest(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Students;
Greatest
--------
SKYLER
```

Voir aussi

«LEAST», à la page 147 détermine la valeur minimale d'une liste.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

LEAST

Cette fonction scalaire (Scalar) renvoie la valeur la plus faible d'une liste d'expressions.

Syntaxe

```
LEAST( value, value [, value ... ] )
```

Paramètre

- value

  Valeur à utiliser pour la comparaison. Toutes les valeurs qui suivent la première valeur sont converties dans le type de données de cette dernière.

Type de valeur renvoyée

Même type de données que l'argument.

Exemple

```
SELECT Least(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Students;
Least
------
SCHOLAR
```

Voir aussi

«GREATEST», à la page 146 détermine la valeur maximale d'une liste.

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

LOWER

Cette fonction scalaire (Scalar) convertit tous les caractères en majuscules en une chaîne en minuscules.

Syntaxe

```
LOWER( string )
```

Paramètre

- string

  Chaîne à convertir.

Type de valeur renvoyée

VARCHAR

Exemple

LOWER('Stage Right') renvoie ‘stage right'.

Voir aussi

«UPPER», à la page 176 convertit tous les caractères en majuscules.

LPAD

Cette fonction scalaire (Scalar) insère une ou plusieurs instances d'une chaîne au début d'une autre chaîne.

Syntaxe

```
LPAD( string, length, [ padChar ] )
```

Paramètres

- string

  Caractère ou chaîne à modifier.
- length

  Longueur d'affichage de la chaîne renvoyée. Cette valeur doit être supérieure ou égale à zéro (0). Si vous utilisez des caractères multi-octets, la longueur correspond au nombre de caractères qui s'affichent ou s'impriment, et non au nombre de multi-octets.
- padChar

  Caractère ou chaîne à insérer. La valeur par défaut est un espace unique ou un caractère vide (' ').

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie une chaîne présentant le même type de caractère que le paramètre string.

Si la longueur spécifiée est inférieure à celle de la chaîne de départ, la fonction renvoie la chaîne tronquée.

Exemples

LPAD('ABC',6,'x') renvoie 'xxxABC'.

LPAD('ABC',6,'xo') renvoie 'xoxABC'.

LPAD('ABC',4) renvoie ' ABC'.

LPAD('ABC', 2, 'x') renvoie 'AB'.

Voir aussi

«RPAD», à la page 173 ajoute des caractères à la fin d'une chaîne.

LTRIM

Cette fonction scalaire (Scalar) supprime des caractères du début d'une chaîne.

Syntaxe

```
LTRIM( wholeString [, setString ] )
```

Paramètres

- wholeString
  Chaîne à tronquer.
- setString
  Caractères à supprimer ; la valeur par défaut est un espace vide unique (' ').

Type de valeur renvoyée

VARCHAR

Remarques

Supprime de façon récursive toutes les instances de setString à partir du début de wholeString jusqu'à ce que wholeString ne commence plus par setString, puis renvoie le résultat correspondant.

Exemples

LTRIM(' ZZZ') renvoie 'ZZZ'.

LTRIM('aaaZZZ','a') renvoie 'ZZZ'.

LTRIM('ababaZZZ','ab') renvoie 'aZZZ'.

LTRIM('abcabaZZZ','abc') renvoie 'abaZZZ'.

Voir aussi

«RTRIM», à la page 174 supprime des caractères de la fin d'une chaîne.

POSITION

Cette fonction scalaire (Scalar) renvoie la position d'un caractère ou d'une chaîne dans une chaîne.

Syntaxe

```
POSITION( sourceForString , searchInString )
```

Variante : POSITION(sourceForString IN searchInString)

Paramètres

- sourceForString
  Chaîne à rechercher.
- sourceInString
  Résultat de l'expression de chaîne dans lequel rechercher.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie la position, en commençant par 1, de la première instance de sourceForString dans le résultat sourceInString. Lorsque CHARACTER_LENGTH(sourceForString) présente la valeur zéro (0), la fonction renvoie 1. Lorsque l'un des arguments présente la valeur NULL, la fonction renvoie le résultat NULL.

Exemples

POSITION('a' IN 'banana') renvoie 2.

POSITION('ana' IN 'banana') renvoie 2.

POSITION('A' IN 'banana') renvoie 0.

POSITION('M' IN 'banana') renvoie 0.

POSITION('' IN 'banana') renvoie 1.

POSITION(NULL IN 'banana') renvoie NULL.

Voir aussi

«SUBSTRING», à la page 175 extrait un caractère ou une sous-chaîne d'une chaîne.

PRIOR_VALUE

Cette fonction scalaire (Scalar) renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.

Syntaxe

```
PRIOR_VALUE( columnName )
```

Paramètre

- columnName

  Colonne ou alias d'un type de données quelconque à évaluer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie une valeur NULL si aucune valeur précédente n'existe, c'est-à-dire lors de la première exécution de la fonction sur la valeur columnName spécifiée.

Utilisez la fonction PRIOR_VALUE() lorsque les données (flux de données) entrent dans le système de façon groupée et triée.

Cette fonction n'est pas autorisée dans la «Clause WHERE», à la page 343 d'une définition de vue.

Exemple

Examinez la requête ci-après identifiant, sous la forme d'un pourcentage d'une heure, la durée requise pour l'exécution d'une tâche, en fonction du nombre de minutes écoulées depuis l'exécution de la tâche précédente :

```
SELECT Task, CAST(
      (TIMESTAMP_DIFF(PRIOR_VALUE(Completed),Completed,
MINUTE ) /60
                   , DECIMAL(5,4)
                  ) AS Hours
    FROM Tasks_Completed
Task             Hours
---------------  ------
Startup
Initialize       0.0887
Begin job        0.1012
Finish job       4.3243
Clean up         0.2500
Shut down        0.1285
Have milk shake  0.6667
```

Voir aussi

«PREV», à la page 187 renvoie une valeur de l'avant-dernière ligne d'un ensemble.

RPAD

Cette fonction scalaire (Scalar) ajoute une ou plusieurs instances d'une chaîne à la fin d'une autre chaîne.

Syntaxe

```
RPAD( string, length, [ padChar ] )
```

Paramètres

- string

  Caractère ou chaîne à modifier.
- length

  Longueur d'affichage de la chaîne renvoyée. Si vous utilisez des caractères multi-octets, la longueur correspond au nombre de caractères qui s'affichent ou s'impriment, et non au nombre de multi-octets.
- padChar

  Caractère ou chaîne à ajouter. La valeur par défaut est un espace unique ou un caractère vide (' ').

Type de valeur renvoyée

VARCHAR

Remarques

Si la longueur spécifiée est inférieure à celle de la chaîne de départ, la fonction renvoie la chaîne tronquée.

Exemples

RPAD('ABC', 6, 'x') renvoie 'ABCxxx'.

RPAD('ABC', 6, 'xo') renvoie 'ABCxox'.

RPAD('ABC', 4) renvoie 'ABC '.

RPAD('ABC', 2, 'x') renvoie 'AB'.

Voir aussi

«LPAD», à la page 170 insère des caractères au début d'une chaîne.

RTRIM

Cette fonction scalaire (Scalar) supprime des caractères de la fin d'une chaîne.

Syntaxe

```
RTRIM( sourceString [, setString ] )
```

Paramètres

- sourceString

  Chaîne à tronquer.
- setString

  Caractères à supprimer ; la valeur par défaut est un espace vide unique (' ').

Type de valeur renvoyée

VARCHAR

Remarques

Supprime de façon récursive toutes les instances de setString à partir de la fin de sourceString jusqu'à ce que sourceString ne finisse plus par setString, puis renvoie le résultat correspondant.

Exemples

RTRIM('ZZZ ') renvoie 'ZZZ'.

RTRIM('ZZZaaa', 'a') renvoie 'ZZZ'.

RTRIM('ZZZababab', 'ab') renvoie 'ZZZ'.

RTRIM('ZZZababc', 'abc') renvoie 'ZZZab'.

Voir aussi

«LTRIM», à la page 171 supprime des caractères du début d'une chaîne.

SUBSTRING

Cette fonction scalaire (Scalar) renvoie une sous-chaîne d'une chaîne spécifiée.

Syntaxe

```
SUBSTRING( string, position, [ length ] )
.
```

Variantes :

```
SUBSTR( string, position, [ length ] )
SUBSTRING( string FROM position [ FOR length ]  )
```

Paramètres

- string

  Chaîne de caractères à rechercher.
- position

  Position de départ de la sous-chaîne, où 1 correspond au premier caractère du début de la chaîne et -1 au dernier caractère. Les valeurs négatives recherchent la sous-chaîne en partant de la fin de la chaîne. La valeur zéro (0) renvoie le même résultat que la valeur 1.
- length

  Longueur de la sous-chaîne à extraire. Si vous n'indiquez aucune longueur, la fonction renvoie tous les caractères situés entre la position spécifiée et la fin de la chaîne. L'indication d'une longueur supérieure au nombre de caractères restants de la chaîne renvoie tous les caractères situés entre la position définie et la fin de la chaîne et remplit la différence avec des caractères d'espacement pour atteindre la longueur spécifiée. Une valeur négative ou égale à zéro (0) renvoie une chaîne vide.

Type de valeur renvoyée

VARCHAR

Exemples

`SUBSTR('breakfast', 1)` renvoie 'breakfast'.

`SUBSTR('breakfast', 0)` renvoie 'breakfast'.

`SUBSTR('breakfast', 30)` renvoie '' (chaîne vide).

`SUBSTR('breakfast', 1, 2)` renvoie 'br'.

`SUBSTR('breakfast', CHARACTER_LENGTH('breakfast'), 2)` renvoie 't'.

`SUBSTR('breakfast', 3, 4)` renvoie 'eakf'.

`SUBSTR('breakfast', 3, 8)` renvoie 'breakfast'.

`SUBSTR('breakfast', -5, 4)` renvoie 'kfas'.

`SUBSTR('breakfast', 1, -1)` renvoie '' (chaîne vide).

Voir aussi

«CHAR_LENGTH», à la page 167 renvoie la longueur d'une chaîne de caractères.

«POSITION», à la page 172 recherche un caractère dans une chaîne.

UPPER

Cette fonction scalaire (Scalar) convertit tous les caractères en minuscules en une chaîne en majuscules.

Syntaxe

```
UPPER( string )
```

Paramètre

- string
  Chaîne de caractères (VARCHAR) à convertir.

Type de valeur renvoyée

VARCHAR

Exemple

UPPER('Volta') renvoie "VOLTA".

Voir aussi

«LOWER», à la page 170 convertit tous les caractères en minuscules.

Séries chronologiques et agrégation

Les fonctions chronologiques et d'agrégation d'ensembles sont destinées aux opérations impliquant des ensembles de fenêtres mobiles et des ensembles de nombres.

Voir aussi «MOV_function», à la page 136.

AVG

Cette fonction d'ensembles (Set) renvoie la valeur moyenne (moyenne arithmétique) d'un ensemble de valeurs numériques.

Syntaxe

```
AVG( numeric )
```

Paramètre

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric.

Remarques

Calcule la moyenne de l'argument numeric dans toutes les lignes de la vue référencée. En cas d'utilisation d'une clause GROUP BY (voir «Clause GROUP BY», à la page 344), la moyenne s'applique à l'argument numeric de chacun des groupes.

```
SELECT AVG( pr_price ) "Average price" FROM Products
Average price
-------------
49.96
```

Exemple

L'exemple ci-après utilise des moyennes mobiles pour produire des résultats semblables à un indicateur MACD (Moving Average Convergence/Divergence, convergence/divergence de la moyenne mobile). (Cet exemple ne constitue pas un indicateur MACD proprement dit, car il n'utilise pas une moyenne mobile exponentielle.) Dans le domaine des transactions de valeurs mobilières, la règle MACD fondamentale consiste à vendre lorsque le MACD tombe en deçà de sa moyenne calculée sur neuf jours et d'acheter lorsqu'il dépasse cette moyenne. Vous pouvez obtenir ce résultat en définissant des règles semblables à ce qui suit :

- Raise SELL lorsque MACD > Nine_Day_MA Lower SELL lorsque MACD < Nine_Day_MA
- Raise BUY lorsque MACD < Nine_Day_MA Lower BUY lorsque MACD > Nine_Day_MA

Pour obtenir ces valeurs, vous devez disposer de deux vues :

- MACD_Base_View répertorie les moyennes mobiles pour chacun des symboles de valeur mobilière dans le flux de données. Notez que la formule Nine_Day_MA répète les formules pour les deux autres moyennes, car vous ne pouvez pas faire référence à un alias d'une autre colonne de la même vue.

```
SELECT
    StockQuotes.SYMBOL AS Symbol,
    MOV_AVG(StockQuotes_DataStream.CLOSE, Day, 26, StockQuotes.DATE)
AS Twentysix_Day_MA,
    MOV_AVG(StockQuotes_DataStream.CLOSE, Day, 12, StockQuotes.DATE)
AS Twelve_Day_MA,
    MOV_AVG((MOV_AVG(StockQuotes.CLOSE, Day, 12, StockQuotes.DATE)-
             MOV_AVG(StockQuotes.CLOSE, Day, 26, StockQuotes.DATE)),
            Day, 9, StockQuotes.DATE ) AS Nine_Day_MA
    FROM StockQuotes
    GROUP BY StockQuotes.SYMBOL
```

- MACD_View contient les dernières valeurs MACD pour chacune des valeurs mobilières stockées dans la vue de base :

```
SELECT MACD_Base_View.Symbol AS Symbol,
       MACD_Base_View.Nine_Day_MA AS Nine_Day_MA,
       (MACD_Base_View.Twentysix_Day_MA -
        MACD_Base_View.Twelve_Day_MA
       ) AS MACD
    FROM MACD_Base_View
```

Voir aussi

«median», à la page 211 renvoie la valeur médiane (centrale) d'un ensemble.

«mode», à la page 212 renvoie la valeur la plus fréquente d'un ensemble.

«MOV_AVG», à la page 181 renvoie la moyenne mobile d'un ensemble.

«TUMBLE_AVG», à la page 189 renvoie la moyenne en défilement continu d'un ensemble.

COUNT

Cette fonction d'ensembles (Set) renvoie le nombre de lignes d'une vue ou d'un ensemble.

Syntaxe

```
COUNT( * )
```

Type de valeur renvoyée

INTEGER

Remarques

Renvoie la valeur zéro (0) si la vue ou l'ensemble est vide.

Cette fonction est également connue sous le nom "count étoile".

Les lignes contenant des valeurs NULL sont comptabilisées.

CURRENT

Cette fonction d'ensembles (Set) renvoie une valeur de la ligne la plus récente ou de la dernière ligne d'un ensemble.

Syntaxe

```
CURRENT( columnName )
```

Paramètre

- columnName

  Colonne ou alias à récupérer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie une valeur de la ligne la plus récente de l'ensemble basée sur l'horodatage de flux de données. Si toutes les lignes de l'ensemble présentent le même horodatage, cette fonction renvoie la valeur de la dernière ligne de l'ensemble.

Sémantique pour les ensembles mobiles

Cette fonction n'est pas utilisable avec un ensemble mobile ou en défilement continu.

Exemple

Rassembler toutes les soumissions de titres et les regrouper par symbole de titre. La ligne "en cours" est toujours la dernière ligne reçue et contient donc le cours acheteur actuel :

```
SELECT symbol, CURRENT(bid) AS Bid,MAX(bid)
AS
High, MIN(bid) AS LOW
    FROM Stock_feed GROUP BY symbol
symbol  Bid    High   Low
------- ------ ------ -----
K       31.25  31.28  30.72
IBM     80.79  80.04  82.55
VCLR    22.60  24.42  22.00
```

Voir aussi

«PREV», à la page 187 renvoie une valeur de la ligne précédant la ligne en cours.

MAX

Cette fonction d'ensembles (Set) renvoie la valeur maximale d'un ensemble.

Syntaxe

```
MAX( expression )
```

Paramètre

- expression

  Expression évaluée sous la forme de n'importe quel type de données et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, ‘z’ est supérieur à ‘A’.

Sémantique pour les ensembles mobiles

Lorsque cette fonction est utilisée sous la forme d'une fonction MOV (voir «MOV_function», à la page 136), elle renvoie la valeur maximale de l'ensemble mobile.

```
MOV_MAX( numeric, window, size [,timestampColumn] )
```

Exemple

Renvoyer le prix maximal de toutes les lignes de l'ensemble Sales :

```
SELECT MAX( price ) FROM Sales;
PRICE
---------
770.00
```

Voir aussi

«MOV_MAX», à la page 183 renvoie la valeur maximale d'un ensemble de fenêtres mobiles.

«TUMBLE_MAX», à la page 191 renvoie la valeur maximale d'un ensemble de fenêtres en défilement continu.

«MIN» renvoie la valeur minimale d'un ensemble.

«GREATEST», à la page 146 renvoie la valeur maximale d'une liste.

MIN

Cette fonction d'ensembles (Set) renvoie la valeur minimale d'un ensemble.

Syntaxe

```
MIN( expression )
```

Paramètre

- expression

Expression évaluée sous la forme de n'importe quel type de données et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, ‘z’ est supérieur à ‘A’.

Sémantique pour les ensembles mobiles

Lorsque cette fonction est utilisée sous la forme d'une «MOV_function», à la page 136, elle renvoie la valeur minimale de l'ensemble mobile.

```
MOV_MIN( numeric, window, size [,timestampColumn] )
```

Exemple

Renvoyer le prix minimal de toutes les lignes de l'ensemble Sales :

```
SELECT MIN( price ) FROM Sales;
PRICE
---------
100.00
```

Voir aussi

«MOV_MIN», à la page 184 renvoie la valeur minimale d'un ensemble de fenêtres mobiles.

«TUMBLE_MIN», à la page 193 renvoie la valeur minimale d'un ensemble de fenêtres en défilement continu.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

«LEAST», à la page 147 renvoie la valeur minimale d'une liste.

MOV_AVG

Cette fonction d'ensembles mobiles (Moving set) renvoie la valeur moyenne (moyenne arithmétique) mobile d'un ensemble de valeurs numériques d'une fenêtre mobile.

Syntaxe

```
MOV_AVG( numeric, window, size [,timestampColumn] )
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.

- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric.

Remarques

Renvoie une valeur NULL si le groupe est vide.

Exemple

Renvoyer le prix moyen de tous les flux de données survenant au cours d'une période de sept jours :

```
MOV_AVG(price, DAY, 7, trade_time) AS Avg_7_day_price
```

Voir aussi

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

«AVG», à la page 177 renvoie la moyenne arithmétique d'un ensemble.

«TUMBLE_AVG», à la page 189 renvoie la moyenne en défilement continu d'un ensemble.

MOV_COUNT

Cette fonction d'ensembles mobiles (Moving set) renvoie le nombre de lignes d'un ensemble d'une fenêtre mobile.

Syntaxe

```
MOV_COUNT( *, window, size [,timestampColumn] )
```

Paramètres

- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

(Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie la valeur zéro (0) si l'ensemble est vide.

Les lignes contenant des valeurs NULL sont comptabilisées.

Lorsque cette fonction est utilisée avec une clause GROUP BY, elle renvoie le nombre de lignes de l'ensemble d'un groupe. Pour en savoir davantage, voir «Interactions GROUP BY», à la page 140.

Exemple

Renvoyer le nombre de tous les événements survenant lors de l'intervalle de 8 heures en cours :

```
MOV_COUNT(*, HOUR, 8, trade_time) AS Total
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
COUNT(*) AS Total OVER (ORDER BY trade_time '8' HOUR)
```

Cette expression est l'équivalent de l'expression ci-dessous après remplissage de toutes les valeurs par défaut :

```
COUNT(*) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING
    REFERENCE OPERATOR)
```

La fenêtre de huit heures débute lorsque le premier événement arrive dans la vue. Pour faire commencer la fenêtre au début de l'heure, indiquez INITIALIZE '2003-03-05 00:00:00.000'.

Voir aussi

La section «Types de fonction», à la page 125 présente les ensembles mobiles.

«COUNT», à la page 178 renvoie le nombre de lignes d'une vue ou d'un ensemble.

«TUMBLE_COUNT», à la page 190 renvoie le nombre de lignes d'un ensemble de fenêtres en défilement continu.

MOV_MAX

Cette fonction d'ensembles mobiles (Moving set) renvoie la valeur maximale d'un ensemble d'une fenêtre mobile.

Syntaxe

```
MOV_MAX( numeric, window, size [,timestampColumn] )
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, 'z' est supérieur à 'A'.

Exemple

Renvoyer le prix maximal de tous les flux de données survenant au cours d'une période de sept jours :

```
MOV_MAX(price, DAY, 7, trade_time) AS Max_7_day_price
```

Voir aussi

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble de fenêtres mobiles.

«TUMBLE_MAX», à la page 191 renvoie la valeur maximale d'un ensemble de fenêtres en défilement continu.

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble.

«GREATEST», à la page 146 renvoie la valeur maximale d'une liste.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

MOV_MIN

Cette fonction d'ensembles mobiles (Moving set) renvoie la valeur minimale d'un ensemble d'une fenêtre mobile.

Syntaxe

```
MOV_MIN( numeric, window, size [,timestampColumn] )
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, 'z' est supérieur à 'A'.

Exemple

Renvoyer le prix minimal de tous les flux de données survenant au cours d'une période de sept jours :

```
MOV_MIN(price, DAY, 7, trade_time) AS Min_7_day_price
```

Voir aussi

«MIN», à la page 180 renvoie la valeur minimale d'une vue ou d'un ensemble.

«TUMBLE_MIN», à la page 193 renvoie la valeur minimale d'un ensemble de fenêtres en défilement continu.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

«LEAST», à la page 147 renvoie la valeur minimale d'une liste.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

MOV_SUM

Cette fonction d'ensembles mobiles (Moving set) renvoie la somme d'un ensemble de valeurs numériques d'une fenêtre mobile.

Syntaxe

```
MOV_SUM( numeric, window, size [,timestampColumn] )
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

«Types de données C-SQL pour les valeurs numériques», à la page 71, même type de données que l'argument numeric. Renvoie une valeur NULL si l'ensemble est vide.

Exemple

Calculer le prix total de tous les événements survenant au cours de l'heure actuelle :

```
MOV_SUM(Price, HOUR, 1)
```

L'instruction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
SUM(Price) OVER (RANGE '1' HOUR PRECEDING
                 REFERENCE OPERATOR)
```

Voir aussi

«SUM», à la page 188 renvoie la somme d'une vue ou d'un ensemble.

«TUMBLE_SUM», à la page 194 renvoie la somme d'un ensemble de fenêtres en défilement continu.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

PREV

Cette fonction d'ensembles (Set) renvoie une valeur de l'avant-dernière ligne d'un ensemble.

Syntaxe

```
PREV( columnName )
```

Paramètre

- columnName

  Colonne ou alias d'un type de données quelconque à évaluer.

Type de valeur renvoyée

Même type de données que l'argument.

Remarques

Renvoie les valeurs de la ligne précédant la ligne "en cours" d'un ensemble, où la ligne en cours correspond à la ligne la plus récente de l'ensemble basée sur l'horodatage d'événement ou, si l'horodatage est le même pour toutes les lignes, à la dernière ligne de l'ensemble.

Sémantique pour les ensembles mobiles

Cette fonction n'est pas utilisable avec un ensemble mobile ou en défilement continu.

Exemple

Rassembler tous les cours de clôture des titres et les regrouper par symbole de titre. La ligne "en cours" est toujours la dernière ligne reçue et contient donc le cours de clôture actuel. La ligne précédente indique la clôture du jour précédent :

```
SELECT symbol, CURRENT(close) AS "Last Trade",
PREV(close) AS "Prev
Cls",
        (CURRENT(close) - PREV(close))
AS Change
    FROM Stock_feed
    GROUP BY symbol
symbol  Last Trade  Prev Cls  Change
------- ----------  --------  ------
K       31.25       31.28     -0.03
IBM     80.79       80.04      0.75
VCLR    24.42       22.60      1.82
```

Voir aussi

«CURRENT», à la page 179 renvoie la valeur de la ligne la plus récente ou de la dernière ligne d'un ensemble.

«PRIOR_VALUE», à la page 148 renvoie la valeur antérieure d'une colonne, d'un alias ou d'une expression.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

SUM

Cette fonction d'ensembles (Set) renvoie la somme d'un ensemble de valeurs numériques.

Syntaxe

```
SUM( numeric )
```

Paramètre

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric. Renvoie une valeur NULL si l'ensemble est vide.

Sémantique pour les ensembles mobiles

Lorsque cette fonction est utilisée sous la forme d'une fonction MOV_ (voir «MOV_function», à la page 136), elle renvoie la somme mobile de l'ensemble mobile.

```
MOV_SUM( numeric, windowwindow, size [,timestampColumn])
```

Dernière valeur de l'ensemble

Lorsque la taille de l'ensemble mobile correspond à un événement unique, la fonction MOV_SUM() conserve la somme des prix des dernières commandes de chacun des clients ayant passé une commande depuis le démarrage du système.

```
SELECT os.os_cust_id, MOV_SUM(os.os_price,
EVENT, 1)
    FROM order_status os
    GROUP BY os.os_cust_id
```

Exemple

Calculer le total de la colonne invOnHand pour toutes les lignes de la table stock :

```
SELECT SUM(invOnHand) "Total on hand"
    FROM stock;
Total on hand
-------------
2
```

Voir aussi

«MOV_SUM», à la page 186 renvoie la somme d'un ensemble de fenêtres mobiles.

«TUMBLE_SUM», à la page 194 renvoie la somme d'un ensemble de fenêtres en défilement continu.

TUMBLE_AVG

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie la valeur moyenne (moyenne arithmétique) d'un ensemble de fenêtres en défilement continu.

Syntaxe

```
TUMBLE_AVG( numeric, window, size [,timestampColumn])
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric.

Remarques

Renvoie une valeur NULL si le groupe est vide.

Exemple

Renvoyer le prix moyen de tous les flux de données survenant au cours d'une période de sept jours :

```
TUMBLE_AVG(price, DAY, 7, trade_time) AS Avg_7_day_price
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
AVG(price) AS Avg_7_day_price OVER (
    ORDER BY trade_time RANGE INTERVAL '7' DAY PRECEDING
SLIDE)
```

Pour déterminer le prix moyen des sept jours précédents, sans inclure le jour en cours, utilisez une fenêtre plutôt que la fonction TUMBLE_AVG(), comme ci-après :

```
AVG(price) AS Avg_prev_7_day_price OVER (
    ORDER BY trade_time
    RANGE BETWEEN INTERVAL '8' DAY PRECEDING
```

```
      AND INTERVAL '1' DAY PRECEDING
SLIDE INTERVAL '7' DAY
INITIALIZE TIMESTAMP '1963-02-18 00:00:00.000'
REFERENCE OPERATOR)
```

Voir aussi

«AVG», à la page 177 renvoie la moyenne arithmétique d'un ensemble.

«MOV_AVG», à la page 181 renvoie la moyenne d'un ensemble de fenêtres mobiles.

La section «Fenêtres en défilement continu», à la page 406 présente les ensembles d'une fenêtre en défilement continu.

TUMBLE_COUNT

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie le nombre de lignes d'un ensemble de fenêtres en défilement continu.

Syntaxe

```
TUMBLE_COUNT( *, window, size [,timestampColumn] )
```

Paramètres

- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie la valeur zéro (0) si l'ensemble est vide.

Si vous utilisez une fenêtre EVENT, cette fonction renvoie un nombre entier inférieur ou égal à la valeur de la fenêtre.

Les lignes contenant des valeurs NULL sont comptabilisées.

Lorsque cette fonction est utilisée avec une clause GROUP BY, elle renvoie le nombre de lignes de l'ensemble d'un groupe.

Exemple

Renvoyer le nombre de tous les événements survenant au cours d'un intervalle de 8 heures :

```
TUMBLE_COUNT(*, HOUR, 8, trade_time) AS Total
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
COUNT(*) AS Total OVER (ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING SLIDE)
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
COUNT(*) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING
    SLIDE INTERVAL '8' HOUR
    REFERENCE OPERATOR)
```

La fenêtre de huit heures débute lorsque le premier événement arrive dans la vue. Pour faire commencer la fenêtre au début de l'heure, indiquez INITIALIZE '2003-03-05 00:00:00.000'.

Voir aussi

«COUNT», à la page 178 renvoie le nombre de lignes d'une vue ou d'un ensemble.

«MOV_COUNT», à la page 182 renvoie le nombre de lignes d'un ensemble de fenêtres en défilement continu.

TUMBLE_MAX

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie la valeur maximale d'un ensemble de fenêtres en défilement continu.

Syntaxe

```
TUMBLE_MAX( numeric, window, size [,timestampColumn])
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la

plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, 'z' est supérieur à 'A'.

Exemple

Renvoyer le prix maximal de tous les événements survenant au cours d'un intervalle d'une heure :

```
TUMBLE_MAX(price, HOUR, 1, trade_time) AS Max_price
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
MAX(price) AS Max_price OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
MAX(price) AS Max_price OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

La fenêtre d'une heure débute lorsque le premier flux de données arrive dans la vue. Pour faire commencer la fenêtre au début de l'heure, indiquez INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'.

De la même façon, la fonction TUMBLE_MAX(price, EVENT, 5) constitue la forme abrégée de la fenêtre complète suivante :

```
MAX(price) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

Voir aussi

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble de fenêtres mobiles.

«MOV_MAX», à la page 183 renvoie la valeur maximale d'un ensemble de fenêtres en défilement continu.

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble.

«GREATEST», à la page 146 renvoie la valeur maximale d'une liste.

La section «Fenêtres en défilement continu», à la page 406 présente les ensembles d'une fenêtre en défilement continu.

TUMBLE_MIN

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie la valeur minimale d'un ensemble de fenêtres en défilement continu.

Syntaxe

```
TUMBLE_MIN( numeric, window, size [,timestampColumn]
)
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

Même type de données que l'argument expression.

Remarques

Dans le cas d'une valeur booléenne, la valeur True est supérieure à la valeur False.

Dans le cas d'une chaîne, 'z' est supérieur à 'A'.

Exemple

Renvoyer le prix minimal de tous les événements survenant au cours d'un intervalle d'une heure :

```
TUMBLE_MIN(price, HOUR, 1, trade_time) AS Min_price
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
MIN(price) AS Min_price OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
MIN(price) AS Min_price OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

La fenêtre d'une heure débute lorsque le premier flux de données arrive dans la vue. Pour faire commencer la fenêtre au début de l'heure, indiquez INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'.

De la même façon, la fonction TUMBLE_MIN(price, EVENT, 5) constitue la forme abrégée de la fenêtre complète suivante :

```
MIN(price) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

Voir aussi

«MIN», à la page 180 renvoie la valeur minimale d'un ensemble de fenêtres mobiles.

«MOV_MIN», à la page 184 renvoie la valeur minimale d'un ensemble de fenêtres en défilement continu.

«MAX», à la page 179 renvoie la valeur maximale d'un ensemble.

«LEAST», à la page 147 renvoie la valeur minimale d'une liste.

La section «Fenêtres en défilement continu», à la page 406 présente les ensembles d'une fenêtre en défilement continu.

TUMBLE_SUM

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie la somme d'un ensemble de valeurs numériques d'une fenêtre en défilement continu.

Syntaxe

```
TUMBLE_SUM( numeric, window, size [,timestampColumn])
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

(Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'utilisez pas cette option si vous souhaitez utiliser le premier flux de données de la vue en tant que base.

Type de valeur renvoyée

Numérique, même type de données que l'argument numeric. Renvoie une valeur NULL si l'ensemble est vide.

Exemple

Cette expression de somme en défilement continu totalise tous les flux de données survenant au cours d'un intervalle d'une heure :

```
TUMBLE_SUM(price, HOUR, 1, trade_time) AS Total
```

La fonction ci-dessus constitue une forme abrégée de l'expression de fenêtre en ligne suivante :

```
SUM(price) AS Total OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SUM(price) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

La fenêtre d'une heure débute lorsque le premier flux de données arrive dans la vue. Pour faire commencer la fenêtre au début de l'heure, indiquez INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'.

De même, la fonction TUMBLE_SUM(price, EVENT, 5) est un raccourci de cette fenêtre complète :

```
SUM(price) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

Voir aussi

«SUM», à la page 188 renvoie la somme d'une vue ou d'un ensemble.

«MOV_SUM», à la page 186 renvoie la somme d'un ensemble de fenêtres mobiles.

La section «Fenêtres en défilement continu», à la page 406 présente les ensembles d'une fenêtre en défilement continu.

Séries chronologiques et statistiques

Les fonctions chronologiques et statistiques sont destinées pour les opérations impliquant des statistiques et des ensembles de fenêtres mobiles basées sur des nombres.

Voir aussi «MOV_function», à la page 136.

MOV_STD_DEVIATION

Cette fonction d'ensembles mobiles (Moving set) renvoie un exemple d'écart type d'un ensemble de valeurs d'une fenêtre mobile.

Syntaxe

```
MOV_STD_DEVIATION( numeric, window, size [,timestampColumn])
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur 0 s'il n'existe qu'une ligne d'entrée. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sqrt{\sum\left[\frac{(X_i - X)^2}{n}\right]}$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemple

Renvoyer l'écart type relatif aux salaires pour les flux de données de l'année passée :

```
SELECT MOV_STD_DEVIATION(salary, YEAR, 1) AS "Dev. for
the last year"
    FROM employees;
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SELECT STD_DEVIATION(salary) AS "Dev. for the last year"
OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    REFERENCE OPERATOR)
```

Voir aussi

«STD_DEVIATION», à la page 202 renvoie l'écart type d'une vue ou d'un ensemble.

«TUMBLE_STD_DEVIATION», à la page 204 renvoie l'écart type d'un ensemble de fenêtres en défilement continu.

«VARIANCE», à la page 207 renvoie le carré de l'écart type.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

MOV_VARIANCE

Cette fonction d'ensembles mobiles (Moving set) renvoie le carré de l'exemple d'écart type d'un ensemble de valeurs d'une fenêtre mobile.

Syntaxe

```
MOV_STD_DEVIATION( numeric, window, size [,timestampColumn]
)
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur zéro (0) si l'ensemble de l'expression ne contient qu'un seul élément. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sum\left[\frac{(X_i - X)^2}{n}\right]$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemple

Renvoyer l'écart relatif aux salaires pour chaque année civile :

```
SELECT MOV_STD_DEVIATION(salary, YEAR, 1) AS "Variation
for last year"
    FROM employees;
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SELECT VARIATION(salary) AS "Variation for the last year"
OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    REFERENCE OPERATOR)
```

Voir aussi

«VARIANCE», à la page 207 renvoie la variance d'une vue ou d'un ensemble.

«STD_DEVIATION», à la page 202 renvoie un écart type.

«TUMBLE_VARIANCE», à la page 206 renvoie la variance d'un ensemble de fenêtres en défilement continu.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

NTILE

Cette fonction de classement (Rank) détermine le rang du niveau de chacune des valeurs d'un ensemble par rapport à l'intégralité de l'ensemble.

Syntaxe

```
NTILE( toRank, tiers )
```

Paramètres

- toRank

  Expression d'un type de données quelconque, faisant généralement référence à une colonne de valeurs à classer.
- tiers

Nombre de niveaux pour la partition des résultats ; nombre entier supérieur à zéro (0). Lorsque cette valeur est supérieure au nombre d'éléments à classer, le même rang est attribué à tous les éléments.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie, pour chacune des lignes de l'ensemble, un nombre entier représentant le niveau auquel appartient la ligne, où un (1) constitue le niveau le plus élevé contenant la valeur maximale. Lorsque le paramètre toRank présente une valeur NULL, celle-ci reçoit le rang le plus bas.

Vous ne pouvez pas utiliser cette fonction en tant qu'argument dans une fonction d'ensembles (Set), une fonction d'ensembles mobiles (Moving set) ou une fonction de classement (Rank). Par exemple, l'instruction SUM(NTILE()) est incorrecte.

Une fonction ntile classe les lignes en tentant de les répartir uniformément au sein d'un ensemble de niveaux fixe. Par exemple, lorsqu'un ensemble de six résultats d'expression {D, B, E, C, A et B} doit être classé en deux niveaux, NTILE() attribue à chacun des niveaux le rang 1 (pour C, D et E) ou 2 (pour A, B et B). Les résultats présentant une valeur identique sont toujours placés dans le même niveau.

Lorsqu'un ensemble de valeurs n'est pas divisible par le nombre de niveaux, la fonction répartit uniformément toutes les lignes en surplus entre les groupes du niveau le plus élevé. Par exemple, le tableau ci-après démontre le mode de répartition des éléments lettres en fonction de différents nombres de niveaux :

Tableau 59. Mode de répartition des éléments lettres en fonction de différents nombres de niveaux

Niveaux :	**1**	**2**	**3**	**4**	**5**	**6**	**7**
A	1	2	3	3	4	5	1
B	1	2	2	2	3	4	1
B	1	2	2	2	3	4	1
C	1	1	2	1	2	3	1
D	1	1	1	1	1	2	1
E	1	1	1	1	1	1	1

Exemple

La requête ci-après classe les ventes de produits de café et de thé en six niveaux selon le rang des ventes. Le classement comportant six niveaux, chacun des noms de produit reçoit une valeur comprise entre 1 et 6. Cet exemple requiert une seule entrée par produit :

```
SELECT prod_name, NTILE( dollars, 6) AS sales_rank
    FROM (lineitem INNER JOIN product
          ON lineitem.item_id=product.productid)
    WHERE product.classkey IN (1, 2, 4, 5);
PROD_NAME           SALES_RANK
Demitasse M         1
```

```
Xalapa Lapa         1
Cafe Au Lait        2
Aroma Roma          2
Veracruzano         3
Colombiano          3
Darjeeling Special  4
Irish Breakfast     4
English Breakfast   5
Earl Grey           5
Gold Tips           6
```

Voir aussi

«RANK» classe les lignes dans l'intégralité de l'ensemble.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

RANK

Cette fonction de classement (Rank) détermine le rang de chacune des valeurs d'un ensemble par rapport à l'intégralité de l'ensemble.

Syntaxe

```
RANK( expression )
```

Paramètre

- expression

  Expression d'un type de données quelconque, faisant généralement référence à une colonne.

Type de valeur renvoyée

INTEGER

Remarques

Renvoie, pour chacune des lignes de l'ensemble, un nombre entier correspondant au rang de la ligne dans l'intégralité de l'ensemble, où la valeur la plus élevée reçoit le rang 1. Lorsque l'expression présente une valeur NULL, celle-ci est classée en dernière position dans la liste de résultats. Par exemple, le classement des valeurs de l'ensemble (10, NULL, 20) attribue le rang 2 à la valeur 10, le rang 1 à la valeur 20 et le rang 3 à la valeur NULL.

Lorsque les valeurs à classer sont identiques, elles reçoivent le même rang, et le rang suivant est omis. Par exemple, les valeurs 4.5, 4.5, 1.0 reçoivent les rangs 1, 1 et 3.

Vous ne pouvez pas utiliser cette fonction en tant qu'argument dans une fonction d'ensembles (Set), une fonction de classement (Rank) ou une fonction d'ensembles mobiles (Moving set). Par exemple, l'instruction SUM(RANK(...)) est incorrecte. Vous ne pouvez pas non plus utiliser une fonction RANK() dans une vue sans état.

Lorsque l'ensemble ne contient qu'une ligne, la fonction RANK() renvoie la valeur 1. Par exemple, RANK(SUM(sales)) = 1.

Exemples

Classer les ventes de produits par région :

```
SELECT RANK( SUM( sales ) ) AS R, SUM(sales) AS S, region
    FROM product_orders
    GROUP BY region
R   S        region
1   100000   north
2   50000    south
```

Classer les ventes de produits par produit :

```
SELECT prod_name, SUM(dollars) AS prod_sales,
RANK( SUM(dollars) ) AS prod_rank
    FROM product, lineitem
    WHERE lineitem.classkey = product.classkey
        AND lineitem.prodkey = product.prodkey
    GROUP BY prod_name;
PROD_NAME      PROD_SALES  PROD_RANK
Demitasse Ms   656401.50   1
Xalapa Lapa    577450.00   2
Aroma Roma     479330.25   5
Verona         467234.00   6
NA Lite        557655.00   3
Lotta Latte    533454.50   4
```

Voir aussi

«NTILE», à la page 198 classe les lignes et les place dans un ensemble de niveaux fini.

La section «Types de fonction», à la page 125 présente les fonctions pour ensembles mobiles.

RATIO_TO_REPORT

Cette fonction de classement (Rank) calcule le rapport d'une valeur à la somme des valeurs pour l'intégralité de l'ensemble.

Syntaxe

```
RATIO_TO_REPORT( numeric )
```

Paramètre

- numeric

  Expression d'un type de données numérique quelconque, faisant généralement référence à une colonne numérique.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie, pour chacune des lignes de l'ensemble, un nombre indiquant le rapport de la ligne à la somme de l'intégralité de l'ensemble. Lorsque l'expression présente une valeur NULL, la fonction renvoie un résultat NULL. Lorsque la somme de l'ensemble est égale à zéro (0), le rapport est également de zéro.

Vous ne pouvez pas utiliser cette fonction en tant qu'argument dans une fonction d'ensembles (Set), une fonction d'ensembles mobiles (Moving set) ou une fonction de classement (Rank). Par exemple, SUM(RATIO_TO_REPORT(...)) est incorrecte.

Exemple

Déterminer le pourcentage que représente chacune des ventes de produit par rapport aux ventes totales de tous les produits sur les 20 dernières semaines :

```
SELECT prod_description DESC,
        SUM(dollars) as sales,
RATIO_TO_REPORT( SUM(
li_amount )) * 100 AS ratio_dollars
    FROM lineitem, product
    WHERE lineitem.li_prod_id = product.prod_id
    GROUP BY prod_description;
```

DESC	SALES	RATIO_DOLLARS
Widget	896931.15	12.88
Basket	514830.00	7.39
Football	507022.35	7.28
Oil Drum	503493.10	7.23
Computer	437863.00	6.29
Chair	429637.75	6.17
Desk	424215.00	6.09
Mesh Bag	421205.75	6.05
Shoelace	417261.00	5.99
Powder	397102.50	5.70
Telephone	394086.50	5.66
Cord	392377.75	5.64
Mouse	389378.25	5.59
Monitor	305859.75	4.39
Case	294982.75	4.24
Cup	236772.75	3.40

STD_DEVIATION

Cette fonction d'ensembles (Set) renvoie un exemple d'écart type d'un ensemble de nombres.

Syntaxe

```
STD_DEVIATION( number )
```

Paramètre

- number

Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur 0 s'il n'existe qu'une ligne d'entrée. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sqrt{\sum\left[\frac{(X_i - X)^2}{n}\right]}$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemple

```
SELECT STD_DEVIATION(salary) "Deviation"
    FROM employees;
Deviation
----------
3909.36575
```

Voir aussi

«MOV_STD_DEVIATION», à la page 196 renvoie l'écart type d'un ensemble de fenêtres mobiles.

«TUMBLE_STD_DEVIATION», à la page 204 renvoie l'écart type d'un ensemble de fenêtres en défilement continu.

«VARIANCE», à la page 207 renvoie le carré de l'écart type.

SUM_OVER_GROUPS

Fonction de classement (Rank) à laquelle sont transmis deux paramètres. une valeur numérique suivie d'un paramètre de classement. Lorsque les groupes sont triés selon ce paramètre de classement, la fonction renvoie une somme évolutive des valeurs numériques.

Syntaxe

```
SUM_OVER_GROUPS([numeric value],[ordering parameter])
```

Paramètres

- numeric value
  Valeurs à additionner.
- ordering parameter
  Classement des valeurs.

Type de valeur renvoyée

Somme évolutive des valeurs numériques fournies dans les arguments.

Exemple

Examinez l'ensemble de données ci-après :

```
grp  data
----------
1    1
2    3
3    10
4    200
```

L'exécution de la fonction suivante :

```
SELECT grp, SUM_OVER_GROUPS(data, grp) as running_sum
GROUP BY grp
```

Renvoie le résultat ci-après :

```
grp  running_sum
----------
1    1
2    4
3    14
4    214
```

TUMBLE_STD_DEVIATION

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie un exemple d'écart type d'un ensemble de valeurs d'une fenêtre en défilement continu.

Syntaxe

```
TUMBLE_STD_DEVIATION( numeric, window, size [,timestampColumn]
)
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit correspondre à un entier positif.
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur 0 s'il n'existe qu'une ligne d'entrée. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sqrt{\sum\left[\frac{(X_i - X)^2}{n}\right]}$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemple

Renvoyer l'écart type relatif aux salaires pour chaque année civile :

```
SELECT TUMBLE_STD_DEVIATION(salary, YEAR, 1) AS "Deviation
per year"
    FROM employees;
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SELECT STD_DEVIATION(salary) AS "Deviation per year"
OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    SLIDE INTERVAL '1' YEAR
    REFERENCE OPERATOR)
```

Notez que vous pouvez utiliser la commande INITIALIZE pour déclarer un exercice fiscal. Envisagez également d'utiliser PARTITION BY pour obtenir les écarts relatifs aux différents échelons de salaire. Par exemple :

```
SELECT STD_DEVIATION(salary) AS "Deviation per year"
OVER (
    PARTITION BY pay_grade
    RANGE INTERVAL '1' YEAR PRECEDING
    SLIDE INTERVAL '1' YEAR
    INITIALIZE '1963-07-01 00:00:00.000'
    REFERENCE OPERATOR)
```

Voir aussi

«STD_DEVIATION», à la page 202 renvoie l'écart type d'une vue ou d'un ensemble.

«MOV_STD_DEVIATION», à la page 196 renvoie l'écart type d'un ensemble de fenêtres en défilement continu.

«VARIANCE», à la page 207 renvoie le carré de l'écart type.

La section «Fenêtres en défilement continu», à la page 406 présente les ensembles d'une fenêtre en défilement continu.

TUMBLE_VARIANCE

Cette fonction d'ensembles en défilement continu (Tumbling set) renvoie le carré de l'exemple d'écart type d'un ensemble de valeurs d'une fenêtre en défilement continu.

Syntaxe

```
TUMBLE_VARIANCE( numeric, window, size [,timestampColumn]
)
```

Paramètres

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue. Pour connaître les autres restrictions, reportez-vous aux descriptions des fonctions concernées.
- window

  Détermine si la taille correspond au nombre d'événements de l'ensemble (EVENT) ou à une durée (SECOND, MINUTE, HOUR, DAY, MONTH ou YEAR).
- size

  Durée ou décompte de la fenêtre à utiliser pour déterminer la taille de l'ensemble. Cette valeur doit être un nombre entier supérieur à zéro (0).
- timestampColumn

  (Facultatif) Valeur de la zone à utiliser comme point de départ de la série chronologique. Ce calcul repose sur la plage de temps à partir de la valeur la plus récente de la colonne dans n'importe quelle ligne de la vue. N'indiquez pas cette option si vous souhaitez baser les calculs sur l'horloge système.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur zéro (0) si l'ensemble de l'expression ne contient qu'un seul élément. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sum\left[\frac{(X_i - X)^2}{n}\right]$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemples

Renvoyer l'écart relatif aux salaires pour chaque année civile :

```
SELECT TUMBLE_VARIATION(salary, YEAR, 1)
        AS "Variation for last year"
    FROM employees;
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SELECT VARIANCE(price) AS "Variation for last year"
```

```
OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' YEAR PRECEDING
    SLIDE INTERVAL '1' YEAR
    REFERENCE OPERATOR)
FROM employees;
```

Voir aussi

«MOV_VARIANCE», à la page 197 renvoie la variance d'un ensemble de fenêtres mobiles.

«STD_DEVIATION», à la page 202 renvoie un écart type.

«VARIANCE» renvoie la variance d'une vue ou d'un ensemble.

La section «Fenêtres en défilement continu», à la page 406 présente les fonctions pour ensembles mobiles.

VARIANCE

Cette fonction d'ensembles (Set) renvoie le carré de l'exemple d'écart type d'un ensemble de valeurs.

Syntaxe

```
VARIANCE( numeric )
```

Paramètre

- numeric

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la valeur zéro (0) si l'ensemble de l'expression ne contient qu'un seul élément. Renvoie une valeur NULL si l'ensemble est vide.

Le résultat est calculé à l'aide de la formule $\sum\left[\frac{(X_i - X)^2}{n}\right]$, où *n* correspond au nombre d'éléments de l'exemple et *X* à l'exemple de moyenne.

Exemple

```
SELECT VARIANCE(salary) "Variance"
    FROM employees;
Variance
----------
15283140.5
```

Voir aussi

«MOV_VARIANCE», à la page 197 renvoie la variance d'un ensemble de fenêtres mobiles.

«STD_DEVIATION», à la page 202 renvoie un écart type.

«TUMBLE_VARIANCE», à la page 206 renvoie la variance d'un ensemble de fenêtres en défilement continu.

Fonctions fournies comme exemples de fonctions UDF

Ces fonctions sont fournies en tant qu'exemples de fonctions définies par l'utilisateur (UDF).

Pour utiliser l'une de ces fonctions, vous devez la charger à partir du répertoire "manifest" dans l'emplacement des programmes de Cognos Real-time Monitoring. Le chemin d'accès est semblable à `realtime\webcontent\sdk\udf\jar\com\cognos\obi\manifest`. Pour plus d'informations, voir Chapitre 35, «Fonctions définies par l'utilisateur», à la page 359.

concatList

Cette fonction scalaire (Scalar) renvoie une chaîne correspondant à la concaténation d'une liste de caractères ou de chaînes.

Syntaxe

```
concatList( string1, string2 [, ... stringN ] )
```

Paramètre

- string

  Expression évaluée sous la forme d'une valeur VARCHAR.

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie la valeur string2 ajoutée à la fin de string1, la valeur string3 ajoutée derrière string2, etc.

Ignore les valeurs NULL, sauf si toutes les valeurs sont NULL, auquel cas la fonction renvoie une chaîne vide.

Exemples

concatList('a','b','c') renvoie 'abc'.

Voir aussi

«CONCAT», à la page 168 renvoie une chaîne correspondant à la concaténation de deux caractères ou chaînes.

«concatSet», à la page 209 renvoie un ensemble de chaînes trié dans l'ordre alphabétique.

La section «Opérateurs de chaîne», à la page 291 décrit l'opérateur `||`.

concatSet

Renvoie un ensemble de chaînes trié dans l'ordre alphabétique.

Syntaxe

```
concatSet( stringExp )
```

Paramètre

- stringExp

  Expression évaluée sous la forme d'une valeur VARCHAR. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

VARCHAR

Remarques

Renvoie une chaîne correspondant à l'ensemble trié de toutes les chaînes transmises à la fonction.

Ignore les valeurs NULL, sauf si toutes les valeurs sont NULL, auquel cas la fonction renvoie une chaîne vide.

Exemples

Examinez l'instruction ci-après :

```
SELECT concatSet(item) AS item_list FROM GroceryList
```

Si les articles de GroceryList sont présentés dans l'ordre suivant :

```
'banana'
'egg'
'apple'
'donut'
NULL
'carrot'
```

Les articles répertoriés par item_list dans la nouvelle vue seront présentés dans l'ordre suivant :

```
'apple,banana,carrot,donut,egg'
```

Par la suite, si vous ajoutez l'article ‘bagel’ à GroceryList, la nouvelle vue affichera les articles dans l'ordre suivant :

```
'apple,bagel,banana,carrot,donut,egg'
```

Voir aussi

«CONCAT», à la page 168 renvoie une chaîne correspondant à la concaténation de deux caractères ou chaînes.

«concatList», à la page 208 renvoie une chaîne correspondant à la concaténation d'une liste de caractères ou de chaînes.

La section «Opérateurs de chaîne», à la page 291 décrit l'opérateur ||.

gammaDist

Cette fonction scalaire (Scalar) renvoie la distribution gamma d'une valeur.

Syntaxe

```
gammaDist( number, alphaNumber, betaNumber, isCumulative)
```

Paramètre

- number : nombre positif à évaluer. Peut être zéro (0).
- alphaNumber : paramètre alpha (nombre positif pouvant être zéro) pour l'équation de distribution gamma.
- betaNumber : paramètre bêta (nombre positif pouvant êtrezéro) pour l'équation de distribution gamma.
- isCumulative : paramètre booléen déterminant la forme de la fonction du nombre reposant sur les paramètres alphaNumber et betaNumber :
  - La valeur TRUE utilise la fonction de distribution cumulée.
  - La valeur FALSE utilise la fonction de masse de probabilité.

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Lorsque le paramètre alphaNumber est égal à un (1), la fonction renvoie une distribution exponentielle.

Lorsque le paramètre alphaNumber est un nombre entier positif, le résultat est une distribution d'Erlang.

Exemple

Remplir une vue avec la masse de probabilité pour une distribution gamma où alpha=9 et bêta=2 :

```
SELECT TimeToFail,
       gammaDist(TimeToFail, 9, 2, FALSE) AS GammaDist
    FROM UnitTests
    ORDER BY TimeToFail
```

Voir aussi

«logNormDist» renvoie la distribution log-normale cumulée d'une valeur.

logNormDist

Cette fonction scalaire (Scalar) renvoie la distribution log-normale cumulée d'une valeur.

Syntaxe

```
logNormDist( number, meanNumber, stdNumber )
```

Paramètres

- number
  Valeur à évaluer.
- meanNumber
  Moyenne arithmétique d'ln(number).
- stdNumber
  Ecart type d'ln(number).

Type de valeur renvoyée

DOUBLE PRECISION

Remarques

Renvoie la distribution log-normale cumulée d'une valeur, où ln(number) est distribué normalement avec une moyenne et un écart type.

median

Cette fonction d'ensembles (Set) renvoie la valeur médiane (centrale) d'un ensemble.

Syntaxe

```
median( numericExp )
```

Paramètre

- numericExp
  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

INTEGER lorsque tous les résultats de numericExp sont des nombres entiers ; DOUBLE PRECISION lorsque l'un des résultats est une valeur décimale.

Remarques

Cette fonction trie les valeurs de l'ensemble, puis renvoie la valeur médiane de l'ensemble trié.

Lorsque le nombre de valeurs est impair, la valeur médiane correspond au nombre central de l'ensemble. Par exemple, la valeur médiane de 2,1,5 est 2, car il s'agit de la valeur centrale de l'ensemble trié.

Si le nombre de valeurs est pair, la valeur médiane correspond à la moyenne des deux nombres centraux de l'ensemble. Par exemple, la valeur médiane de 2,1,5,4 est 3, car il s'agit de la moyenne des valeurs 2 et 4. En outre, si le résultat de la moyenne est une valeur décimale, ce résultat est arrondi au nombre entier inférieur : la valeur médiane de 2 et de 3 est 2, qui correspond au résultat de l'instruction floor(2.5).

Cette fonction ignore les valeurs NULL.

Exemples

Examinez l'instruction ci-après :

```
SELECT median(Value) AS MedianV FROM NumberList
```

Le résultat de cette instruction est 1.5 (moyenne des valeurs 1.0 et 2.0 après omission des valeurs NULL) lorsque les éléments de NumberList sont présentés dans l'ordre suivant :

```
3.0
NULL
0.0
2.0
1.0
NULL
NULL
```

Le résultat de l'instruction est 1 si l'ensemble est le suivant :

```
1
2
```

Les valeurs de l'ensemble étant des nombres entiers, le résultat doit également être un nombre entier. Par conséquent, la valeur moyenne de l'ensemble, correspondant à 1.5, est arrondie au nombre entier inférieur : 1.

Voir aussi

«AVG», à la page 177 renvoie la moyenne arithmétique d'un ensemble.

«mode» renvoie la valeur la plus fréquente d'un ensemble.

mode

Cette fonction d'ensembles (Set) renvoie la valeur la plus fréquente d'un ensemble.

Syntaxe

```
mode( numericExp )
```

Paramètre

- numericExp

  Expression évaluée sous la forme d'une valeur numérique et ne pouvant pas faire référence à une fonction de classement. L'argument correspond généralement à l'une des colonnes d'une vue.

Type de valeur renvoyée

Même type de données que le résultat numericExp.

Remarques

Lorsque plusieurs valeurs surviennent à la même fréquence, la fonction mode() renvoie la première valeur rencontrée. Reportez-vous aux exemples.

Cette fonction ignore les valeurs NULL.

Exemples

Examinez l'ensemble de valeurs ci-après indiquées dans la fonction mode() dans l'ordre suivant :

```
1
3
4
1
3
```

La fonction mode() renvoie la valeur 1, car cette dernière survient le plus fréquemment et a été rencontrée avant la valeur 3 qui survient à la même fréquence. Si l'ensemble avait été spécifié dans l'ordre inverse, la fonction mode() aurait renvoyé la valeur 3.

Pour l'ensemble de valeurs ci-après, la fonction mode renvoie le résultat 1.0 :

```
1
3
NULL
2.0
NULL
NULL
1
```

Les valeurs NULL sont ignorées, et la valeur 2.0 contraint la fonction mode() à renvoyer une valeur DOUBLE PRECISION.

Voir aussi

«AVG», à la page 177 renvoie la moyenne arithmétique d'un ensemble.

«median», à la page 211 renvoie la valeur médiane (centrale) d'un ensemble.

yield

Cette fonction calcule le rendement relatif à un ensemble de résultats de test (échec=0, réussite=1), c'est-à-dire le nombre d'unités ayant réussi le test par rapport au nombre total d'unités. Les valeurs Null sont ignorées.

Syntaxe

```
yield(pass, unit)
```

Paramètres

- pass

  Indique si le rendement doit être calculé en fonction du nombre d'unités ayant réussi le test ou du nombre d'unités ayant échoué. Indiquez la valeur pass si vous souhaitez baser le calcul du rendement sur le nombre d'unités ayant réussi le test. Indiquez la valeur fail si vous souhaitez baser le calcul du rendement sur le nombre d'unités ayant échoué au test.
- unit

  Indique le nombre total d'unités à tester.

Type de valeur renvoyée

Renvoie la valeur NULL si le rendement est inférieur à zéro ; dans le cas contraire, la fonction renvoie une valeur DOUBLE PRECISION.

Exemple

Si les valeurs ci-après sont transmises à la fonction SELECT YIELD(pass, 4) FROM E :

0, 1 1, 0 0, 1 NULL, NULL

La fonction renvoie le résultat 0.5 car le test 1 et le test 3 ont réussi, mais le test 2 a échoué. Le test 4 est ignoré.

Si vous transmettez les mêmes valeurs à la fonction SELECT YIELD(fail, 4) FROM E, la fonction renvoie le résultat 0.25 car un seul des quatre tests a réussi.

Chapitre 16. HTTP Post

Une table de flux de données HTTP Post reçoit des événements issus d'une action d'envoi HTTP, soit en tant que résultat d'un formulaire HTML envoyé par un navigateur, soit sous forme de données codées dans une URL connectée à la table.

Fonctionnement de HTTP Post

Les données d'événements HTTP Post arrivent incorporées dans une URL. L'agent interne extrait les zones de l'URL, puis insère les données dans la table de flux de données.

Le Post HTTP peut être le résultat de la soumission d'un formulaire HTML contenant des zones <INPUT> par un utilisateur, ou être créé par une application communiquant dans le protocole HTTP.

Figure 17. Table de flux de données recevant des données

Tables de flux de données HTTP Post

Une table de flux de données HTTP Post reçoit de nouveaux événements à partir d'une action d'envoi HTTP, qui est généralement le résultat d'un formulaire HTTP envoyé depuis un navigateur.

Dans un formulaire HTML, chaque élément <INPUT> est mappé à une colonne dans la table de flux de données. Les données d'événement peuvent également être publiées dans l'adresse URL qui transmet les zones au système. Vous trouverez des exemples dans la section «Envoi vers une table de flux de données HTTP Post», à la page 218.

Pour créer une table de flux de données HTTP Post, vous devez disposer d'une autorisation de création sur les tables. (Pour plus d'informations sur ces autorisations, voir «Autorisation de création d'objets», à la page 299.) Le tableau ci-dessous dresse la liste des attributs d'une table de flux de données HTTP Post.

© Copyright IBM Corp. 2007, 2014

Tableau 60. Attributs des tables de flux de données HTTP Post

Attribut	Description
Nom	Nom de la table de flux de données. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Description	Description facultative qui peut contenir tous les caractères de texte.
Enregistrer dans	Indique le dossier dans lequel la table de flux de données sera stockée. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Envoi vers l'adresse URL	Adresse URL à laquelle envoyer les informations données. Vous trouverez des exemples dans la section «Envoi vers une table de flux de données HTTP Post», à la page 218.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Si les événements peuvent être traités dans n'importe quel ordre, désactivez cet attribut. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive l'événement lorsque le nombre d'erreurs consécutives défini est atteint. Par exemple, si la valeur définie est 5, le flux de données est désactivé au bout de 5 erreurs consécutives. Toutefois, dans le cas où 4 erreurs se produisent, puis 0 erreur, puis 2 erreurs, le flux de données reste actif. La valeur par défaut est **Ne pas désactiver**.
Informations sur la colonne	L'onglet **Informations sur la colonne** détermine comment placer les zones HTTP Post dans les colonnes de la table de flux de données. Cette dernière contient une colonne pour chaque zone. Pour en savoir davantage, voir «Informations sur les colonnes HTTP Post», à la page 217.
Intervalle de suppression de l'état	Cet onglet comporte plusieurs options de suppression des données d'événement. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.

Création d'une table de flux de données HTTP Post

Utilisez la procédure ci-après pour créer une table de flux de données HTTP Post.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Flux de données**.
4. Dans la fenêtre Nouveau flux de données, sélectionnez **Flux de données simple**.
5. Sélectionnez **Envoi HTTP** comme type de source de la table de flux de données.
6. Affectez un nom à la table de flux de données et définissez les colonnes de cette dernière dans l'onglet **Informations sur la colonne**.
7. Enregistrez la table d'envoi HTTP avec l'état activé.

Informations sur les colonnes HTTP Post

L'onglet Informations sur la colonne définit comment faire correspondre les zones du message HTTP Post avec les colonnes de la table de flux de données.

Il existe une zone par colonne de la table de flux de données, chacune d'elles comportant les attributs suivants :

Tableau 61. Informations sur les colonnes HTTP Post

Attribut	Description
Nom de zone	Nom de la colonne dans la table de flux de données.
Nom du message	Nom de la zone dans le message. Sur un formulaire HTML, il s'agit de l'attribut NAME affecté à chaque élément de formulaire. Vous trouverez des exemples dans la section «Envoi vers une table de flux de données HTTP Post», à la page 218. Lors du mappage d'une zone Fichier à plat, le nom de chaque zone intégrée est Non disponible et il n'est pas modifiable.
Type de données	Type de données associé à la colonne dans la table de flux de données.
Format	(Facultatif) Format associé à la colonne dans la table de flux de données pour les valeurs VARCHAR (chaîne) et DECIMAL.

Chaque zone du message peut correspondre à une zone simple mappée directement à une colonne de la table de flux de données, ou à une zone plus complexe (zone de fichier à plat) qui comporte plusieurs zones mappées individuellement à des colonnes de la table. Les zones complexes sont traitées en tant que fichiers à plat au format CSV, à largeur fixe ou XML. Pour en savoir davantage, voir Chapitre 13, «Fichiers à plat», à la page 103.

Les zones de message peuvent contenir une ou plusieurs lignes de données. Toutefois, chaque ligne fait partie du même événement, contrairement aux importations de fichier à plat qui traitent chaque ligne du fichier comme un événement unique.

Ajout d'une zone de message

Vous pouvez ajouter une zone de message afin de fournir des informations supplémentaires.

Pour plus de détails sur le type de source, voir :

- «Fichiers à largeur fixe», à la page 110
- «Fichiers délimités», à la page 110
- «Prise en charge des fichiers XML», à la page 111

Pour plus d'informations sur l'envoi de données vers des zones de fichier à plat, voir «Evénements d'entrée multiples», à la page 220.

Pour plus d'informations sur les attributs de fichiers à plat, voir «Tables de flux de données associées aux fichiers à plat», à la page 104.

Procédure

1. Dans l'onglet **Informations sur la colonne**, cliquez sur **Ajouter une zone Fichier à plat**.
2. Choisissez le type de fichier à plat que vous souhaitez associer à la zone de message.

   Pour les fichiers délimités et à largeur fixe, vous pouvez identifier un fichier exemple basé sur des données réelles pour faciliter le mappage des colonnes. Les données de ce fichier sont présentées à l'étape suivante afin de vous guider tout au long du processus de mappage des données d'événement dans la table.
3. Pour les fichiers à largeur fixe, définissez l'emplacement des colonnes de données à l'aide de la boîte de dialogue **Définir les largeurs des zones**.
4. Spécifiez les attributs du fichier à plat.
5. Définissez les informations sur la colonne en fonction du format choisi.
6. Cliquez sur **Enregistrer le flux de données**.

Modification de la définition d'une zone de message

Vous pouvez modifier la définition d'une zone de message.

Procédure

1. Ouvrez un flux de données existant.
2. Cliquez sur **Modifier**.
3. Dans l'onglet **Informations sur la colonne**, sous **Mise en forme**, remplacez la valeur par **<Changer la mise en forme>**.

   Lors de la modification d'une zone de message, l'option de fichier exemple pour les types de fichiers délimités et à largeur fixe n'est pas disponible.

Envoi vers une table de flux de données HTTP Post

La plupart des événements HTTP Post sont générés à partir d'un formulaire HTML. Lorsque vous définissez l'événement, définissez une colonne pour chaque élément <INPUT> nommé.

Examinez l'exemple de formulaire HTML ci-après :

```
<FORM action="http://.../now/postservlet?eventname=Example"
      method="post">
    <P>
    <LABEL for="name">Name: </LABEL>
        <INPUT type="text" name="name"><BR>
    <LABEL for="name">Date (yyyy-mm-dd): </LABEL>
        <INPUT type="text" name="date"><BR>
    <LABEL for="amt">Amount: </LABEL>
        <INPUT type="text" name="amt"><BR>
    <LABEL for="switch">Switch: </LABEL>
        <INPUT type="radio" name="switch" value="FALSE"
checked >Off
        <INPUT type="radio" name="switch" value="TRUE">On<BR>
    <INPUT type="submit" value="Send">
    <INPUT type="reset">
    </P>
</FORM>
```

Les quatre zones de ce formulaire correspondent aux quatre colonnes ci-après de la table de flux de données :

Figure 18. Les quatre zones de ce formulaire correspondent aux quatre colonnes ci-après de la table de flux de données

La zone de date correspond au type de données VARCHAR, et non TIMESTAMP. Dans les vues dérivées de ce flux de données, convertissez ces valeurs en date et heure à l'aide de la commande cast, comme ci-après :

```
CAST(httpDataStream."Date" AS TIMESTAMP)
```

Envoi vers des zones message

Pour transmettre les données dans une zone message, vous pouvez incorporer les informations dans l'URL ou utiliser un élément HTML <TEXTAREA> et entrer les données de fichier à plat dans cette zone.

Par exemple, votre formulaire HTML peut comporter la déclaration suivante :

```
<LABEL for="flatfile">Flat file text: </LABEL>
    <TEXTAREA name="MessageField" rows="20" cols="80"></TEXTAREA>
```

Dans le navigateur, vous pouvez choisir d'entrer les données manuellement ou de copier les données d'un fichier à plat et de les coller manuellement dans la zone. N'oubliez pas que les données doivent être exprimées dans le format de la zone **Fichier à plat** déclarée, tel que le format délimité.

Pour en savoir plus sur l'intégration d'informations dans l'URL, voir «Envoi de valeurs dans l'URL».

Avertissement :

- Lorsque la table de flux de données contient d'autres colonnes que la colonne de la zone message, vous ne pouvez entrer qu'une seule ligne de données dans la zone <TEXTAREA>. Si la colonne de la zone message est la seule colonne, vous pouvez entrer plusieurs lignes de données d'événements.
- Plusieurs lignes transmises par l'intermédiaire d'un élément <TEXTAREA> sont considérées comme faisant partie du même événement, contrairement à un fichier texte dont chacune des lignes correspond à un événement unique.

Envoi de valeurs dans l'URL

Lorsque vous transmettez les valeurs de zone directement dans l'URL, affectez un nom et une valeur à chacune des zones en les séparant par une perluète (&).

Par exemple :

```
...?eventname=Example&name="Skyler"&date=2003-03-05&amt=9.21&switch=TRUE
```

En revanche, si vous transmettez les valeurs à une zone de fichier à plat délimité, nommez la zone et séparez les valeurs à l'aide d'un caractère de séparation (une virgule généralement). Par exemple :

```
...?eventname=Example&msgFile="Skyler",2003-03-05,9.21,TRUE
```

Evénements d'entrée multiples

Pour envoyer plusieurs événements à une zone de fichier à plat, vous pouvez les séparer par %0D%0A (transmission MIME pour une fin de ligne : "CR LF").

Par exemple :

```
...="Skyler",2003-03-05,9.21,TRUE%0D%0A"Mike",1963-02-18,9.01,FALSE
```

Si vous envisagez d'envoyer plusieurs lignes, la zone **Fichier à plat** doit être la seule zone de la liste Colonne de la table de flux de données. Lorsque la liste inclut d'autres colonne, une seule ligne d'entrée est autorisée.

Chapitre 17. Java Messaging Service (JMS)

Java Messaging Service (JMS) permet d'accéder aux messages générés par les applications Java.

L'application de production publie des messages dans les rubriques ou les files d'attente auxquelles l'agent IBM Cognos Real-time Monitoring est abonné. Chaque message publié dans une rubrique ou une file d'attente est associé à un nouvel événement dans la table de flux de données correspondante.

Les agents JMS d'IBM Cognos Real-time Monitoring communiquent avec les classes d'objets de rubrique ou de file d'attente JMS, gérées par les serveurs d'applications Web. Lorsque vous définissez l'agent, vous spécifiez le mode qu'il doit utiliser pour se connecter à la classe d'objets. Lorsque vous définissez une table de flux de données JMS, vous lui demandez de s'abonner à une rubrique ou une file d'attente JMS gérée par la classe d'objets avec laquelle l'agent est en communication. Ensuite, lorsque la rubrique publie un nouveau message ou lorsque la file d'attente reçoit un nouveau message, l'agent le réceptionne et le transmet à la table de flux de données.

Figure 19. Fonctionnement JMS

Les agents Cognos Real-time Monitoring JMS prennent en charge les types de corps JMS MapMessage et TextMessage pour les messages de rubrique et de file d'attente. Le type JMS MapMessage est constitué de paires nom-valeur, dans lesquelles les noms sont des chaînes et les valeurs sont des enveloppes de type Java. Pour en savoir davantage, voir «Types de données JMS (Java Messaging Service)», à la page 224.

Flux de données JMS

Un flux de données JMS reçoit de nouvelles données d'événements d'une application Java qui publie des messages dans la rubrique ou envoie des messages à la file d'attente à laquelle la table est abonnée. Chaque nouveau message de rubrique ou de file d'attente figure comme un nouvel événement dans la table.

Si le message contient une chaîne de caractères complexe au format CSV (délimité) ou à largeur fixe, il est recommandé d'avoir un fichier exemple qui contienne les données au format de la chaîne d'événement. Vous pouvez utiliser cet exemple

© Copyright IBM Corp. 2007, 2014

lorsque vous créez l'événement pour vous assurer que les zones correspondent correctement à la table de flux de données en visualisant l'ajout des lignes de données dans les colonnes.

Le tableau ci-dessous décrit les attributs d'un flux de données JMS.

Attribut	Description
Nom	Identifie la table de flux de données. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Enregistrer dans	Indique le dossier dans lequel l'objet de flux de données sera stocké. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Description	Description facultative qui peut contenir tous les caractères de texte.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Sinon, si les événements peuvent être traités dans n'importe quel ordre, désactivez cette option. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Agent JMS	Agent JMS qui se connecte au flux de messages JMS. Créez un agent File d'attente JMS ou Rubrique JMS à l'aide du bouton **Activités** de Workbench. Pour en savoir davantage, voir «Agents de file d'attente JMS (Java Messaging Service)», à la page 35 ou «Agents de rubrique JMS», à la page 38.

Attribut	Description
Rubrique ou file d'attente	Identifie la rubrique ou la file d'attente à laquelle le message est envoyé et défini par le module de publication de message. Il s'agit d'une adresse JNDI similaire à com.obi.myjmstopic sur Oracle WebLogic et topic/com.obi.myjmstopic sur JBoss.
Sélecteur de message	Expression booléenne qui applique une condition de filtre aux messages envoyés par le module de publication. La syntaxe de cette condition est identique à celle de la clause WHERE de la commande SELECT (voir «Clause WHERE», à la page 343). Ce filtre n'accepte que les messages dans lesquels la propriété Supplier contient l'une des trois valeurs suivantes : Supplier IN ('Xyz, Corp', 'Ink, Inc', 'Gizmos')
Informations sur la colonne	Les zones **Informations sur la colonne** déterminent comment placer les zones du message JMS dans les colonnes de la table de flux de données. Cette dernière contient une colonne pour chaque zone. Pour en savoir davantage, voir «Informations sur la colonne dans Java Messaging Service», à la page 224.
Intervalle de suppression de l'état	Cet onglet contient plusieurs options permettant de supprimer les données d'événements sauvegardées. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.

Remarque : L'agent JMS prend uniquement en charge les types de corps JMS MapMessage et TextMessage. Le type MapMessage est constitué de paires nom-valeur, dans lesquelles les noms sont des chaînes et les valeurs sont des enveloppes de type Java. TextMessage est constitué d'une seule chaîne de texte sans nom, qui peut être interprétée en tant que fichier XML ou en tant que fichier délimité ou à largeur fixe. Pour plus de détails, voir «Types de données JMS (Java Messaging Service)», à la page 224.

Création d'une table de flux de données à partir d'une connexion JMS (Java Messaging Service)

Vous pouvez créer une table de flux de données à partir d'une connexion JMS.

Avant de commencer

Avant de créer un flux de données à partir d'un agent JMS, vous devez disposer d'une autorisation de création sur les tables de correspondance et les flux de données et d'une autorisation en lecture seule sur l'agent qui va alimenter la table. Vous devez également avoir accès à une rubrique ou une file d'attente JMS identifiée par l'emplacement JNDI de la classe d'adresses.

Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Ouvrez l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Flux de données**.
4. Sélectionnez JMS comme type de source.
5. Définissez les valeurs des attributs de la table de flux de données.
6. Dans l'onglet **Informations sur la colonne**, définissez les colonnes de la table de flux de données.
7. Enregistrez le flux de données JMS avec l'état activé.

Informations sur la colonne dans Java Messaging Service

Les zones Informations sur la colonne déterminent comment mapper les zones du message JMS aux colonnes de la table de flux de données. Il y a une zone pour chaque colonne de la table.

Chaque zone du message peut être une zone simple qui s'insère directement dans une colonne d'événement ou une zone complexe (une zone Fichier à plat) constituée de plusieurs zones dont chacune s'insère dans des colonnes de la table. Les zones complexes sont traitées en tant que fichiers à plat au format délimité, à largeur fixe ou XML. Pour obtenir une description détaillée de ces types de fichiers, voir Chapitre 13, «Fichiers à plat», à la page 103.

Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-dessous.

Tableau 62. Attributs de la colonne

Attribut	Description
Nom de zone	Nom de la colonne dans la table de flux de données.
Nom du message	Nom de la zone dans le message. Lorsque vous associez une zone de message, le nom de chaque zone incorporée est Non disponible et il n'est pas modifiable. Lorsque vous attribuez un nom au message, ce dernier représente un message de mappe et son nom est celui du message. Si vous n'attribuez pas de nom au message, le système vous demande de confirmer qu'il s'agit d'une charge utile au message texte JMS lors de la sauvegarde.
Type de données	Type de données de la colonne. Pour en savoir davantage, voir «Types de données JMS (Java Messaging Service)».
Format	Format de la colonne des valeurs VARCHAR (chaîne) et DECIMAL. L'attribut Format est facultatif.

Types de données JMS (Java Messaging Service)

Les types de données des messages JMS associés correspondent aux types de données IBM Cognos Real-time Monitoring.

L'exemple suivant décrit les types de données des messages JMS associés qui correspondent aux types de données Cognos Real-time Monitoring. Pour en savoir davantage, voir Chapitre 9, «Types de données», à la page 69.

Tableau 63. Mappage des types de données des messages

Type de données Java	**Type de données Cognos Real-time Monitoring**	**Type de données Java**	**Type de données Cognos Real-time Monitoring**
boolean	Booléen	long	Long
byte	Entier	flottant	Double
short	Entier	double	Double
char	Varchar	Chaîne	Varchar
int	Entier	octet[]	Non pris en charge

Chapitre 18. Connectivité JDBC (Java Database Connectivity)

JDBC (Java Database Connectivity) est une interface de programmation Java qui permet d'accéder à des bases de données SQL à partir de programmes Java.

IBM Cognos Real-time Monitoring utilise JDBC pour effectuer les actions suivantes :

- Récupérer des données de la table de correspondance à partir d'une base de données relationnelle (SGBDR), tel que décrit dans «Tables JDBC (Java Database Connectivity)» et «Agents JDBC (Java Database Connectivity)», à la page 238.
- Accéder à la base de métadonnées dans laquelle Cognos Real-time Monitoring stocke les définitions d'objet et d'état.
- Autoriser d'autres applications Java à accéder aux données de vue métier stockées en mémoire. Pour plus d'informations, voir Chapitre 19, «Accès JDBC (Java Database Connectivity) aux données des vues et des cubes», à la page 243.

Figure 20. Interfaces JDBC

Tables JDBC (Java Database Connectivity)

Les tables de correspondance et de flux de données JDBC reçoivent leurs données des systèmes SGBDR externes. Ces données sont extraites par l'intermédiaire d'une requête exécutée sur la base de données ou d'une procédure stockée dans le SGBD.

Tables de correspondance

Dans le cadre des tables de correspondance, les nouvelles données ne sont extraites que lorsqu'un nouvel événement le requiert. Ensuite, l'agent transmet les données de la requête au SGBD qui renvoie le résultat.

© Copyright IBM Corp. 2007, 2014

Pour plus d'informations sur les tables de correspondance, voir Chapitre 20, «Tables de correspondance», à la page 265.

Figure 21. Table de correspondance

Exemple de table de correspondance

Prenons l'exemple d'une vue qui relie une table de flux de données à une table de correspondance :

```
SELECT DataStream.ID, Lookup.Name, Lookup.BDate AS Birth_Date
    FROM DataStream, Lookup WHERE DataStream.ID = Lookup.ID
```

La sortie de la table de correspondance utilisée par cette vue peut ressembler à ce qui suit.

```
Field Name      JDBC Data Type   Data Type
------------    --------------   ---------
Name            STRING           Varchar
BDate           DATE             Timestamp
```

Création d'une source JDBC (Java Database Connectivity) pour une table de flux de données

Vous pouvez créer une source JDBC pour une table de flux de données.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Flux de données**.
4. Sélectionnez **Flux de données simple** ou **Flux de données consolidé**.
5. Sélectionnez **JDBC** comme type de source pour le flux de données.
6. Sélectionnez la source **Requête** ou **Procédure stockée**.
   - Une source Procédure stockée appelle une procédure enregistrée dans le SGBD pour localiser les données.
   - Une source Requête crée une requête SQL SELECT sur la base de données dans le langage natif de la base de données. Saisissez l'instruction SELECT

dans la zone de requête. Lorsque vous cliquez sur **Continuer**, Workbench valide la requête et détermine les colonnes renvoyées.

7. Enregistrez la source JDBC avec l'état activé.

Création d'une source JDBC (Java Database Connectivity) pour une table de correspondance

Vous pouvez créer une source JDBC pour une table de correspondance.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Table de correspondance**.
4. Choisissez JDBC comme type de source de la table de correspondance.
5. Cliquez sur **Parcourir** pour sélectionner la connexion JDBC.
6. Choisissez la source **Requête** ou **Procédure stockée**.
   - Une source Procédure stockée appelle une procédure enregistrée dans le SGBD pour localiser les données.
   - Une source Requête crée une requête SQL SELECT sur la base de données dans le langage natif de la base de données. Saisissez l'instruction SELECT dans la zone de requête. Lorsque vous cliquez sur **Continuer**, Workbench valide la requête et détermine les colonnes renvoyées.
7. Enregistrez la source JDBC avec l'état activé.

Tables de flux de données

Concernant les tables de flux de données, l'agent interroge régulièrement le système SGBD pour voir si de nouveaux événements sont disponibles, puis les récupère afin de les inclure dans la table de flux de données. Chaque événement renvoyé est traité de manière individuelle, quel que soit le nombre d'événements retourné par la requête d'interrogation.

Figure 22. Agent JDBC interrogeant un système SGBD à la recherche de nouveaux flux de données

Avant de créer un flux de données vers une table source JDBC, assurez-vous de disposer des éléments suivants :

- Un agent

  Un agent JDBC possédant des droits d'accès suffisants pour interroger la base de données ou appeler la procédure stockée. Pour créer un agent, cliquez sur le bouton **Activités** et choisissez **Créer**, puis **Flux de données** dans les listes déroulantes. Pour en savoir davantage, voir «Agents JDBC (Java Database Connectivity)», à la page 238.

- Pour les requêtes

  Les schémas des tables à interroger.
- Pour les flux de données de requête

  Une colonne de la table source doit contenir une valeur incrémentée qui identifie les nouveaux événements lorsqu'ils sont disponibles. Pour en savoir davantage, voir «Interrogation de la source JDBC (Java Database Connectivity)», à la page 233.
- Pour les procédures stockées

  Pour définir la procédure dans le SGBDR et dresser la liste des zones d'entrée et de sortie (ensemble de résultats), ainsi que leurs types de données. Pour en savoir davantage, voir «Source de procédure stockée», à la page 235.
- Autorisations

  Autorisation de création (voir Chapitre 23, «Autorisations», à la page 295) sur les tables (voir «Autorisation de création d'objets», à la page 299) et autorisation Lecture seule sur l'agent qui alimente la table.

Une table JDBC possède les attributs décrits dans le tableau ci-dessous :

Tableau 64.

Attributs	Description
Nom	Identifie la table. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Description	Description facultative qui peut contenir tous les caractères de texte.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée (flux de données consolidés uniquement)	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Si les événements peuvent être traités dans n'importe quel ordre, désactivez cet attribut. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.

Tableau 64. (suite)

Attributs	Description
Agent JDBC	Agent source JDBC existant autorisé à accéder à un système SGBD relationnel. Pour créer un agent, cliquez sur le bouton **Activités** et choisissez **Créer**, puis **Flux de données** dans les listes déroulantes. Pour plus d'informations sur les agents, voir «Agents JDBC (Java Database Connectivity)» , à la page 238.
Requête JDBC	Instruction SELECT réalisée sur la base de données en langage de base de données natif. Si vous modifiez la requête, cliquez sur **Soumettre à nouveau la requête** pour la valider. Vous ne pouvez pas effectuer d'enregistrement avec une requête non valide. Pour plus d'informations sur la commande SELECT, voir la documentation de référence liée à SGBD.
Désactiver la table de correspondance lorsque ce nombre d'erreurs consécutives	Nombre d'erreurs consécutives pouvant être reçues avant que le système désactive la table de correspondance. Une fois désactivée, une table de correspondance doit être réactivée manuellement.
Traiter toutes les lignes de l'ensemble de résultats comme un seul événement	Toutes les lignes renvoyées dans l'ensemble de résultats sont considérées comme un événement unique. Sinon, chaque ligne renvoyée de la table est considérée comme un événement distinct.
Informations sur la zone	Colonnes à remplir dans la table de correspondance ou de flux de données. Les noms des zones sont issus du résultat de la requête JDBC. Lorsque le système valide la requête, il remplit cette liste de zone et identifie le type de données JDBC de chaque valeur retournée. Vous pouvez indiquer le type de données associé de la colonne de table ainsi que le format du type de données (si applicable). Les noms des zones sont identiques à ceux définis dans le schéma SGBD, sauf si vous avez créé des alias à l'aide de l'opérateur AS dans la liste de sélection de l'instruction SELECT.
Clé d'événement (flux de données uniquement)	Pour les fichiers délimités et à largeur fixe, la clé permet d'identifier les colonnes de zone clé pour les événements multilignes. Pour en savoir davantage, voir «Evénements sur plusieurs lignes», à la page 107.

Tableau 64. (suite)

Attributs	Description
Index (tables de correspondance uniquement)	Crée un index de la colonne, lorsque cette case est cochée. Lorsque l'option prefetch est activée pour la mise en cache des données, il est crucial que vous sélectionniez l'index approprié pour obtenir de bonnes performances. Sélectionnez **Index** pour les colonnes que vous utilisez dans les conditions de jointure. Si vous n'utilisez pas la mise en cache prefetch, l'activation de l'option **Index** n'a aucun effet.
Mise en cache des données (tables de correspondance uniquement)	Stocke les résultats de la requête dans la mémoire. Les futures requêtes extraient des données de la mémoire, ce qui réduit le nombre de requêtes sur le SGBD. Pour en savoir davantage, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.
Interrogation (flux de données uniquement)	Fréquence d'appel de la procédure stockée ou de recherche de nouveaux événements dans le SGBD. Pour en savoir davantage, voir «Interrogation de la source JDBC (Java Database Connectivity)», à la page 233.
Intervalle de suppression de l'état	Options permettant de supprimer les données d'événements sauvegardées. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.
Upsert (flux de données uniquement)	Options permettant d'activer la fonctionnalité Upsert pour les flux de données JDBC. La fonctionnalité Upsert combine les actions de mise à jour, d'insertion et de suppression de données provenant des tables de flux de données. Pour en savoir davantage, voir «Upsert», à la page 234.

Source de requête

Une source Requête crée une requête SELECT sur la base de données dans le SGBD source.

Les requêtes sont utilisées dans les tables de correspondance et de flux de données, et elles respectent la syntaxe native utilisée par le SGBD. Pour plus d'informations sur la syntaxe, consultez la documentation sur le SGBD.

Limitations Sybase

Lorsque vous envoyez une requête à une base de données Sybase, vous devez tenir compte des limitations suivantes.

- Tous les noms, tables et colonnes incluses, sont sensibles à la casse.
- Toutes les requêtes doivent se présenter sous la forme table SELECT * FROM ; vous ne pouvez inclure aucune clause SELECT. Pour filtrer les résultats, chargez-les dans une vue métier, puis filtrez cette vue.

Exemple de table de flux de données utilisant une requête d'interrogation

Cette rubrique fournit un exemple de table de flux de données qui utilise une requête d'interrogation.

Prenons l'exemple de la requête suivante associée à une table de flux de données.

```
SELECT * FROM sales
```

Les informations sur les zones renvoyées pourraient être celles-ci :

```
Field Name      JDBC Data Type   Data Type
------------    --------------   ---------
Sales_ID        INT              Integer
Product_ID      INT              Integer
Customer_ID     STRING           Varchar
```

Notez que la zone d'incrémentation est très probablement "Sales_ID". Pour plus d'informations, voir «Interrogation de la source JDBC (Java Database Connectivity)».

Interrogation de la source JDBC (Java Database Connectivity)

L'interrogation indique à l'objet la fréquence à laquelle appeler la procédure stockée ou interroger le SGBD à la recherche de nouveaux événements.

L'interrogation possède les paramètres décrits dans le tableau suivant :

Tableau 65. Paramètres d'interrogation de la source JDBC

Paramètre	Description
Interroger selon un intervalle	Fréquence à laquelle appeler la procédure ou émettre la requête.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive l'objet (interrompt l'interrogation) après plusieurs erreurs consécutives. Par défaut, l'interrogation s'arrête après cinq erreurs consécutives. Pour réactiver l'objet, modifiez son état (voir «Etat d'un objet», à la page 273) en activé. Spécifiez la valeur zéro (0) pour que l'objet ne soit jamais désactivé.
Zone d'incrémentation (requêtes de flux de données uniquement)	Identifie la colonne de la table source, qui contient une valeur incrémentée automatiquement à chaque nouvel événement. Par exemple, si la table interrogée contient des ID uniques croissants, cette zone est celle qui est utilisée par la requête qui applique la logique "où l'ID est supérieur à l'ID maximal de la dernière requête".

Tableau 65. Paramètres d'interrogation de la source JDBC (suite)

Paramètre	Description
Valeur initiale (requêtes de flux de données uniquement)	Valeur de départ affectée à la zone d'incrémentation. Cette valeur initiale n'est pas incluse dans le résultat. La première valeur incluse dans le résultat est la valeur supérieure la plus proche à la valeur initiale. Par exemple, si vous indiquez la valeur initiale 500, la première valeur incluse est 501. Pour les requêtes suivantes, cette valeur doit être supérieure à celle de la zone d'incrémentation renvoyée par la dernière requête. Si vous utilisez une base de données de table de correspondance SQLServer 2005 et un pilote Microsoft 2005, les dates initiales doivent se présenter au format yyyy-mm-dd ; sinon, une erreur peut apparaître.

Upsert

La fonctionnalité upsert combine les actions de mise à jour, d'insertion et de suppression de données provenant des tables de flux de données. Vous pouvez activer cette fonctionnalité pour les flux de données JDBC, les flux de fichiers à plat, les flux JMS, etc.

La fonctionnalité upsert met à jour de façon dynamique votre base de données. Lorsque de nouveaux événements sont ajoutés aux tables de base de données, d'anciens événements sont remplacés ou supprimés, et la base de données contient toujours uniquement les données nécessaires.

Pour activer la fonctionnalité upsert, vous avez besoin d'un agent JDBC dédié à upsert. Cet agent gère les données requises pour la fonctionnalité upsert. Vous pouvez créer l'agent JDBC upsert avant ou au moment de création du flux de données. Par défaut, upsert utilise le même agent JDBC que celui spécifié pour la création d'un flux de données JDBC.

Vous activez la fonctionnalité upsert lors de la création du flux de données JDBC. N'activez pas cette fonctionnalité lorsque le flux de données utilise la clé d'événement.

Les paramètres suivants sont associés à cette fonctionnalité :

Tableau 66. Paramètres Upsert

Paramètre	Description
Activer les mises à jour/insertions/ suppressions (Upsert)	Active la fonctionnalité upsert pour un flux de données JDBC.
Agent JDBC	Désigne l'agent JDBC. Vous pouvez accepter l'agent par défaut que vous avez sélectionné pour le flux de données ou cliquer sur **Parcourir** et indiquer un autre agent.

Tableau 66. Paramètres Upsert (suite)

Paramètre	Description
Nom de la table pour le stockage des données upsert	Indique le nom de la table dans laquelle sont stockées les données requises pour la fonctionnalité upsert. Cette table est créée dans la base de données associée à l'agent JDBC. Supprimez cette table si vous supprimez ou désactivez le flux upsert.
Age maximal des données upsert	Indique la période pendant laquelle prendre en charge les données upsert. L'action de mise à jour ou de suppression n'est pas valide pour un événement antérieur à cette période.
Colonne clé	Indique la colonne à utiliser pour identifier les événements à supprimer ou à mettre à jour.
Colonne action	Indique la colonne qui doit contenir les indicateurs "I/D/U" pour les actions d'insertion, de suppression et de mise à jour. Vous pouvez sélectionner plusieurs colonnes. Toutefois, ne sélectionnez pas la colonne déjà définie en tant que **Colonne action**.

Source de procédure stockée

La source de procédure stockée JDBC requiert des attributs spécifiques.

Ces attributs sont décrits dans le tableau ci-dessous :

Tableau 67. Attributs de la source de procédure stockée JDBC

Attribut	Description
Nom de la procédure	Nom de la procédure stockée dans le SGBD. Les flux de données JDBC ne prennent pas en charge un jeu de résultats multiples ou des paramètres de sortie de procédure stockée. Pour Oracle, cela signifie que seules les fonctions Oracle sont prises en charge car elles ne renvoient qu'un seul résultat.
Résultats	Colonnes de la table de correspondance ou de flux de données, avec leurs types de données et, éventuellement, la mise en forme. La procédure renvoie un résultat dont les valeurs correspondent aux colonnes, dans l'ordre où elles s'affichent dans la liste. Le type de données identifie le type de colonne dans la table et est automatiquement converti à partir du type JDBC défini dans «Mappage des types de données JDBC (Java Database Connectivity)» , à la page 237.

Tableau 67. Attributs de la source de procédure stockée JDBC (suite)

Attribut	Description
Entrées	(Facultatif pour les flux de données.) Paramètres transmis à la procédure stockée, avec leurs types de données. Ces paramètres contiennent les valeurs à rechercher dans la table du SGBD. Les entrées sont transmises à la procédure en tant qu'arguments, dans l'ordre où elles s'affichent dans la liste. Le type de données identifie le type de données IBM Cognos Real-time Monitoring de la valeur transmise à la procédure. Pour en savoir davantage, voir «Mappage des types de données JDBC (Java Database Connectivity)», à la page 237. En outre, pour les flux de données, la procédure recherche généralement dans le SGBD les événements insérés depuis le dernier appel de la procédure. En fait, elle identifie dans la table les zones qui contiennent des valeurs d'incrémentation ou d'augmentation. Par exemple, si la table interrogée contient des ID uniques croissants, cette zone est celle qui est utilisée par la procédure, qui applique la logique "où l'ID est supérieur à l'ID maximal de la dernière requête". Pour les flux de données, la **valeur d'interrogation initiale** spécifie la valeur à utiliser lorsque la procédure recherche pour la première fois des événements dans le SGBD. Par exemple, vous pouvez spécifier des ID commençant à 500. Pour les requêtes suivantes, la **valeur d'interrogation ultérieure** identifie une zone contenant la valeur maximum de la dernière requête. Cette valeur est une zone **Sortie** du précédent résultat.
Interrogation	Fréquence d'appel de la procédure stockée. Pour en savoir davantage, voir «Interrogation de la source JDBC (Java Database Connectivity)», à la page 233.

Lorsque vous lancez une requête sur une base de données Sybase, gardez à l'esprit que les noms, y compris ceux des tables et des colonnes, sont sensibles à la casse.

Exemple de réception de données de table de correspondance à l'aide d'une procédure stockée

Vous pouvez consulter un exemple de vue qui relie une table de flux de données à une table de correspondance.

Examinez l'exemple ci-après :

```
SELECT DataStream.ID, Lookup.Name
```

```
    FROM DataStream, Lookup
    WHERE DataStream.ID = Lookup.ID
```

La table de sortie de cette vue est la colonne Name et l'entrée de la procédure est la colonne ID.

```
Output Field Name Data Type
-----------------  ---------
Name               Varchar
Input Field Name Data Type
-----------------  ---------
ID               Integer
```

Exemple de réception d'un événement à l'aide d'une procédure stockée

Vous pouvez consulter un exemple de réception d'un événement à l'aide d'une procédure stockée.

Prenons l'exemple d'un événement avec les zones suivantes. Notez qu'Event_Timestamp est une zone contenant une valeur unique et incrémentée : chaque enregistrement d'événement reçoit une information d'horodatage du SGBD.

```
Output Field Name Data Type
-----------------    ---------
Ticket_ID            Varchar
Cust_ID              Varchar
Status               Varchar
Topic                Integer
When_Opened           Timestamp
Assigned_To          Varchar
Event_Timestamp      Timestamp
```

Le flux de données d'entrée identifie un paramètre qui transmet la valeur à la requête. Le nom de l'entrée doit être unique dans la liste, mais il ne joue aucun autre rôle. Dans l'exemple suivant, le nom de cette zone est "IN1".

```
Input Field Name Initial Polling Value Subsequent Polling Value
----------------  ---------------------  ------------------------
IN1               2003-03-05 19:45:00    DataStream_Timestamp
```

Lorsqu'elle interroge le SGBD pour la première fois, la procédure stockée émet une requête similaire à l'exemple ci-après :

```
SELECT * FROM datastreams
    WHERE datastream_time >= "2003-03-05 19:45:00"
```

Les requêtes suivantes utilisent les résultats de cette requête comme point de départ pour la recherche de nouveaux événements.

Mappage des types de données JDBC (Java Database Connectivity)

Les types de données des colonnes du SGBD sont affichés en tant que types de données JDBC et correspondent aux types de données IBM Cognos Real-time Monitoring.

Pour en savoir davantage, voir Chapitre 9, «Types de données», à la page 69.

Ce mappage est décrit dans le tableau ci-dessous.

Tableau 68. Mappage des types de données JDBC

Type de données JDBC	Caractère	Entier	Double	Decimal	Horodatage	Booléen	Long
CHAR	oui					littéral	
VARCHAR	oui					littéral	
LONGVARCHAR	oui					littéral	
NUMERIC		oui	oui	oui		oui	oui
DECIMAL		oui	oui	oui		oui	oui
BIT		oui	oui	oui		oui	oui
TINYINT		oui	oui	oui		oui	oui
SMALLINT		oui	oui	oui		oui	oui
INTEGER		oui	oui	oui		oui	oui
BIGINT		oui	oui	oui		oui	oui
REAL		oui	oui	oui		oui	oui
FLOAT		oui	oui	oui		oui	oui
DOUBLE		oui	oui	oui		oui	oui
BINARY							
VARBINARY							
DATE					oui		
TIME					oui		
TIMESTAMP					oui		

N'utilisez pas le type de données booléen Cognos Real-time Monitoring dans un prédicat WHERE transmis à la source JDBC. Les valeurs booléennes peuvent être incluses dans la liste Select.

Agents JDBC (Java Database Connectivity)

Un agent JDBC (Java Database Connectivity) communique avec un SGBDR (Système de gestion de base de données relationnelle) par l'intermédiaire d'une requête exécutée sur la base de données ou d'une procédure stockée dans le SGBD. Ce dernier renvoie une ou plusieurs lignes de données, que l'agent transmet aux tables de correspondance ou de flux de données qui en font la demande.

Les agents JDBC sont synchrones, ce qui signifie qu'ils extraient les messages d'événement et les données de table de correspondance suite à l'émission d'une requête, comme décrit dans le tableau ci-après. Pour les tables de correspondance, les agents accèdent au SGBD lorsqu'un nouvel événement requiert des données de table de correspondance. Pour les tables de flux de données, l'agent interroge régulièrement le SGBD, extrait les nouveaux événements détectés et les inclut dans la table de flux de données. Chaque événement renvoyé est traité de manière individuelle, quel que soit le nombre d'événements retourné par la requête d'interrogation.

Tableau 69. Extraction de données des agents JDBC

Envoi de flux de données	Extraction de flux de données	Extraction de table de correspondance
Non	Oui	Oui

Création d'un agent JDBC

Vous pouvez créer un agent JDBC.

Les agents JDBC possèdent les attributs suivants :

Tableau 70. Attributs de l'agent JDBC

Attribut	Description
Nom	Identifie l'agent. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Enregistrer dans	Indique le dossier dans lequel enregistrer l'agent. La valeur par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Description	Description facultative qui peut contenir tous les caractères de texte.
Type de base de données	Identifie le fournisseur SGBD en tant que DB2, Oracle, SQL Server, Sybase, MySQL, PostgresSQL ou Teradata.
Nom d'utilisateur	Nom d'utilisateur à saisir pour se connecter au SGBD. Les droits d'accès en requête sont obligatoires.
Mot de passe	Mot de passe pour le nom d'utilisateur. Si vous oubliez cette option, l'agent utilise le mot de passe spécifié dans la définition de la configuration de la source JDBC du serveur d'applications.
Nombre maximal de lignes par requête	Le nombre maximal de lignes pour retourner le résultat de la requête. Cette option permet d'empêcher les utilisateurs de retourner des résultats trop longs qui peuvent affecter le SGBD.
Type de connexion JDBC	Définit la méthode de connexion au JDBC dans le serveur d'applications. Les méthodes disponibles sont les suivantes : Basée sur la source de données : se connecte à un pool de bases de données JDBC par l'intermédiaire d'une connexion JNDI. Basée sur une adresse URL : se connecte à une source JDBC via une adresse URL.

Tableau 70. Attributs de l'agent JDBC (suite)

Attribut	Description
Nom JNDI de la source JDBC (source de données uniquement)	Nom de la source de données à utiliser en tant que connexion à la base de données. Ce nom est au format JNDI, comme com.cognos.obi.products.ProductSource. Une source obtient sa connexion à partir d'un pool de connexions gérées par le serveur d'applications. Ce serveur laisse les connexions ouvertes afin de réduire les délais lors de l'établissement d'une connexion. Le pools de connexions doit être configuré en tant que pool non transactionnel (non-TxT dans WebLogic).
Adresse URL JDBC (adresse URL uniquement)	URL qui renvoie à la connexion JDBC configurée dans le serveur d'applications exécutant IBM Cognos Real-time Monitoring. Par exemple, ce type d'URL peut se présenter comme suit : jdbc:oracle:thin:some_context/ context@v480:1521:symbols
Classe du pilote JDBC (adresse URL uniquement)	Pilote JDBC à utiliser. Ce pilote doit résider dans le chemin d'accès aux classes de l'application exécutant Cognos Real-time Monitoring. Indiquez le nom complet de la classe, comme oracle.jdbc.driver.OracleDriver, dans le chemin d'accès aux classes.
Propriétés JNDI (source de données uniquement)	Propriétés JNDI (Java naming and directory interface) facultatives et supplémentaires requises pour créer ou garder l'agent dans la source JDBC. Vous pouvez utiliser ces paires nom-valeur pour spécifier les propriétés JDBC. Les noms correspondent à l'un des raccourcis ci-dessous ou à une propriété JNDI reconnue. L'agent reconnaît les noms suivants en tant que raccourcis vers les propriétés JNDI : factory correspond à INITIAL_CONTEXT_FACTORY. provider correspond à PROVIDER_URL. security_credentials correspond à SECURITY_CREDENTIALS. security_principal correspond à SECURITY_PRINCIPAL.
Propriétés de connexion (adresse URL uniquement)	Définit les propriétés de connexion supplémentaires et facultatives nécessaires à la création et la mise à jour de l'agent dans la source JDBC. Vous pouvez utiliser ces paires nom-valeur pour spécifier les propriétés de la connexion JDBC.

Avant de commencer

Avant de créer un agent JDBC, vous devez :

- disposer de l'autorisation de création sur les agents ;
- définir une source de données JDBC gérée par le serveur d'applications, idéalement un serveur qui dispose d'un pool de connexions. Configurez le pool de connexions comme spécifié dans la documentation de votre serveur d'applications. De plus, dans la définition du pool :
  - Définissez un nombre maximum de connexions simultanées à la base de données, supérieur à 200.
  - Configurez une fréquence d'actualisation supérieure à 0, idéalement égale à 1 ou 2 minutes, de manière à arrêter et redémarrer la base de données sans perdre la connexion au pool. De plus, vous pouvez configurer le pool pour qu'il s'assure de l'existence d'une table physique dans la base de données.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Dans la fenêtre Créer un agent, choisissez JDBC comme type de source.
5. Spécifiez les attributs en fonction de vos besoins.
6. Enregistrez l'agent avec l'état activé.

Chapitre 19. Accès JDBC (Java Database Connectivity) aux données des vues et des cubes

IBM Cognos Real-time Monitoring fournit une interface de programmation d'application (API) qui permet aux applications JDBC 2.0 de récupérer les données d'une vue ou d'un cube, et de récupérer les métadonnées qui décrivent les vues ou les cubes de l'installation.

- Vous pouvez visualiser les données d'une vue récente. Si cette vue contient une fenêtre d'ensembles mobiles (Moving set), les données de la fenêtre sont renvoyées.
- En définissant des paramètres de propriété spécifiques dans l'URL JDBC, vous pouvez également afficher les colonnes suivantes :
  - Identificateur d'événement (VC_EVENT_ID), qui identifie l'événement ayant produit la toute dernière ligne insérée dans la vue.
  - Identificateur du dernier événement (VC_LATEST_EVENT_ID), qui identifie le dernier événement ayant provoqué la mise à jour de la vue, même si les données de cet événement ne sont peut-être pas incluses dans la vue.
  - Horodatage de l'événement (VC_TIMESTAMP), qui identifie le moment auquel le dernier événement a été inclus dans la vue.

Pour en savoir davantage, voir «URL JDBC (Java Database Connectivity)».

Chemin d'accès aux classes

Le chemin d'accès aux classes pour le pilote JDBC d'IBM Cognos Real-time Monitoring (cqjdbcclient.jar) et commons-logging-1.1.jar doit être ajouté dans l'application JDBC client.

Le chemin d'accès aux classes est le suivant :

```
java -classpath .;commons-logging_1.1.jar;
cqjdbcclient.jar JDBCAccessor
```

Vous pouvez obtenir le fichier commons-logging-1.1.jar à partir du site Apache.org.

Le fichier cqjdbcclient.jar se trouve dans le répertoire <rép_install_cognos>/ realtime/sdk/java/lib.

URL JDBC (Java Database Connectivity)

L'adresse URL JDBC comporte un format spécifique.

```
jdbc:cognos:obi://<host>:<port>[;<property>=<value>][;<property>=<value>]...
```

Le numéro de port par défaut est 2669. Vous pouvez définir le port JDBC par le biais des paramètres **Contrôle du système** dans **Paramètres du système** de la **Console d'administration**.

Les propriétés et leurs valeurs par défaut sont les suivantes :

© Copyright IBM Corp. 2007, 2014

Tableau 71. Propriétés et valeurs par défaut des adresses URL JDBC

Propriétés	Valeurs par défaut	Description
implicitColumns	faux	Indique si les colonnes implicites sont renvoyées en plus des données de révision normales. Si la valeur est true, les colonnes VC_EVENT_ID, VC_LATEST_EVENT_ID et VC_TIMESTAMP sont incluses dans l'ensemble de résultats.
C8passport		Si IBM Cognos 8 authentifie un utilisateur, ce dernier peut se servir du pilote JDBC en transmettant la propriété C8passport au pilote.
Nom d'utilisateur		Utilisateur de base de données au nom duquel la connexion est établie.
password		Mot de passe de l'utilisateur.

Les incidences des paramètres de propriétés sont les suivantes :

- Si la propriété C8passport est définie, les propriétés du nom d'utilisateur et du mot de passe sont ignorées.
- La propriété du mot de passe est ignorée et l'utilisateur est connecté en tant qu'utilisateur anonyme, si les conditions suivantes sont remplies :
  - La propriété C8passport n'est pas définie.
  - L'accès anonyme est activé.
  - La propriété du nom d'utilisateur est définie sur Anonyme ou elle n'est pas indiquée.
- Si la propriété C8passport n'est pas définie et que l'utilisateur anonyme n'est pas activé, les valeurs fournies pour les propriétés du nom d'utilisateur et du mot de passe sont utilisées.

Si la valeur de la propriété implicitColumns est définie sur true, les données de la vue récente, l'identificateur d'événement, l'identificateur du dernier événement et l'horodatage d'événement sont identiques à celles qui sont indiquées dans une base de données lors de la conservation des vues. Pour en savoir davantage, voir «Conservation des vues dans une base de données», à la page 372.

Interfaces des vues JDBC (Java Database Connectivity)

JDBC 2.0 définit des interfaces permettant d'accéder aux données.

IBM Cognos Real-time Monitoring implémente les interfaces pour accéder aux données.

Pour obtenir une description détaillée sur les interfaces, voir la documentation JDBC à l'adresse suivante http://java.sun.com/j2se/1.4.2/docs/api/java/sql/package-summary.html.

Le pilote JDBC Cognos Real-time Monitoring ne fournit pas de prise en charge complète de l'API JDBC standard. Pour plus d'informations sur l'implémentation de l'API JDBC Real-time Monitoring, voir Annexe A, «Méthodes JDBC (Java Database Connectivity)», à la page 423.

Pour obtenir la liste des classes du pilote JDBC Cognos Real-time Monitoring (Driver, Connections, Statement, DatabaseMetaData, ResultSet et ResultSetMetaData) ainsi que les méthodes associées qu'il fournit, voir Chapitre 19, «Accès JDBC (Java Database Connectivity) aux données des vues et des cubes», à la page 243.

Mappages de types de données

Les fonctions get de cette classe renvoient les valeurs des colonnes IBM Cognos Real-time Monitoring.

Ce tableau indique les fonctions à utiliser pour les différents types de données Real-time Monitoring.

Tableau 72. Mappages de types de données

Type de données	getBoolean	getInt	getLong	get Double	getBigDecimal	getString	getTimestamp
Booléen	Oui	Non	NA	Non	Non	Non	NA
Entier	Non	Oui		Non	Non	NA	NA
Long	NA	Non	Oui	Non	Non	Non	NA
Double	Non	Non		Oui	NA	NA	NA
Décimal	Non	Non		Non	Oui	Non	NA
Varchar	Non	Non		Non	Non	Oui	Non
Horodatage	NA	NA		NA	NA	Non	Oui

La valeur **Oui** indique que la fonction renvoie une valeur compatible avec le type de données Real-time Monitoring.

La valeur **Non** indique des types de données qui peuvent être compatibles, mais dont la conversion n'est pas recommandée.

Récapitulatif des colonnes de la fonction getColumns()

Récapitulatif de la documentation Java relative à l'élément DatabaseMetaData.getColumns().

Pour obtenir la liste complète, voir la documentation Java.

Tableau 73. Récapitulatif des colonnes de la fonction getColumns()

Colonne	Type	Description
TABLE_CAT	Chaîne	Catalogue de table (peut avoir la valeur NULL).
TABLE_SCHEM	Chaîne	Schéma de la table (peut avoir la valeur NULL).
TABLE_NAME	Chaîne	Nom de la table.
COLUMN_NAME	Chaîne	Nom de la colonne.
DATA_TYPE	Entier	Type SQL du fichier java.sql.Types. Voir «Valeurs renvoyées par la colonne DATA_TYPE», à la page 247 pour obtenir un récapitulatif.
TYPE_NAME	Chaîne	Nom du type dépendant de la source de données. Pour un type UDT, le nom doit être complet.

Tableau 73. Récapitulatif des colonnes de la fonction getColumns() (suite)

Colonne	Type	Description
COLUMN_SIZE	Entier	Taille de la colonne. Pour des données de type caractère ou date, correspond au nombre maximal de caractères. Pour des données de type numérique ou décimal, il s'agit de la précision d'arrondi.
BUFFER_LENGTH	Chaîne	Non utilisée.
DECIMAL_DIGITS	Entier	Nombre de fractions.
NUM_PREC_RADIX	Entier	Base (en général 10 ou 2).
NULLABLE	Entier	Valeur Is NULL autorisée. columnNoNulls : peut ne pas autoriser les valeurs NULL. columnNullable : permet définitivement les valeurs NULL. columnNullableUnknown : la possibilité d'attribution des valeurs NULL est inconnue.
REMARKS	Chaîne	Commentaire qui décrit la colonne (peut avoir la valeur NULL).
COLUMN_DEF	Chaîne	Valeur par défaut (peut avoir la valeur NULL).
SQL_DATA_TYPE	Entier	Non utilisée.
SQL_DATETIME_SUB	Entier	Non utilisée.
CHAR_OCTET_LENGTH	Entier	Pour les types de caractères, correspond au nombre maximal d'octets dans la colonne.
ORDINAL_POSITION	Entier	Index de la colonne dans la table (à partir de 1).
IS_NULLABLE	Chaîne	La valeur "NO" signifie que la colonne n'autorise pas les valeurs NULL. La valeur "YES" signifie que la colonne peut les autoriser. Une chaîne vide indique que la possibilité d'attribution des valeurs NULL est inconnue.

Récapitulatif des colonnes de la fonction getTables()

Ce tableau constitue un récapitulatif de la documentation Java pour la fonction DatabaseMetaData.getTables().

Consultez la liste complète des colonnes dans la documentation Java.

Tableau 74. Récapitulatif des colonnes de la fonction getTables()

Colonne	Type	Description
TABLE_CAT	Chaîne	Catalogue de la table (peut avoir la valeur NULL).

Tableau 74. Récapitulatif des colonnes de la fonction getTables() (suite)

Colonne	Type	Description
TABLE_SCHEM	Chaîne	Schéma de la table (peut avoir la valeur NULL).
TABLE_NAME	Chaîne	Nom de la table.
TABLE_TYPE	Chaîne	Type de la table. Les types classiques sont "TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL TEMPORARY", "LOCAL TEMPORARY", "ALIAS", "SYNONYM".
REMARKS	Chaîne	Colonne utilisée par IBM Cognos Real-time Monitoring pour modifier un horodatage.
TYPE_CAT	Chaîne	Catalogue de types (peut avoir la valeur NULL).
TYPE_SHEM	Chaîne	Schéma de types (peut avoir la valeur NULL).
TYPE_NAME	Chaîne	Nom du type (peut avoir la valeur NULL).
SELF_REFERENCING_COL_NAME	Chaîne	Nom de la colonne "identificateur" d'une table basée sur un type structuré (peut avoir la valeur NULL).
REF_GENERATION	Chaîne	Décrit la procédure de création des valeurs dans SELF_REFERENCING_COL_NAME. Les valeurs possibles sont "SYSTEM", "USER", "DERIVED" (peut avoir la valeur NULL).

Valeurs renvoyées par la colonne DATA_TYPE

La colonne DATA_TYPE renvoie une valeur de type nombre entier qui identifie le type de données Java.

Pour plus de détails, voir le fichier java.sql.Types. Les valeurs disponibles sont répertoriées dans le tableau ci-dessous.

Tableau 75. Valeurs renvoyées par la colonne DATA_TYPE

Type	Valeur
BIT	-7
TINYINT	-6
BIGINT	-5
LONGVARBINARY	-4
VARBINARY	-3
BINARY	-2
LONGVARCHAR	-1
NULL	0
CHAR	1
NUMERIC	2
DECIMAL	3
INTEGER	4
SMALLINT	5
FLOAT	6

Tableau 75. Valeurs renvoyées par la colonne DATA_TYPE (suite)

Type	Valeur
REAL	7
DOUBLE	8
VARCHAR	12
DATE	91
TIME	92
TIMESTAMP	93
OTHER	1111
JAVA_OBJECT	2000
DISTINCT	2001
STRUCT	2002
ARRAY	2003
BLOB	2004
CLOB	2005
REF	2006

Exemples d'accesseurs JDBC

Les exemples de cette section montrent comment se connecter aux serveurs IBM Cognos Real-time Monitoring et interroger les données et métadonnées sur les vues et les colonnes définies dans l'installation.

Java

L'accès au pilote JDBC dépend des classes java.sql.*.

Veillez à inclure l'importation suivante dans vos applications :

```
import java.sql.*;
```

Résultat de l'exemple

Pour compiler l'application, utilisez la commande ci-dessous.

```
javac -classpath . JDBCAccessor.java
```

Pour exécuter le programme, chargez-le dans le serveur d'applications.

Les résultats des exemples sont imprimés en sortie standard, et les erreurs sont standard.

Exemple : Etablissement d'une connexion aux serveurs IBM Cognos Real-time Monitoring

Cet exemple montre comment établir une connexion avec le pilote JDBC des serveurs IBM Cognos Real-time Monitoring.

Etablissez la connexion en créant un objet Connection à l'aide de la syntaxe suivante :

```
Connection connection = null;
connection = DriverManager.getConnection(url, userName,password);
```

Les paramètres userName (nom d'utilisateur) et password (mot de passe) identifient un compte Cognos Real-time Monitoring. Lors de l'interrogation d'un objet particulier (par exemple une vue), le compte utilisateur doit disposer au moins d'une autorisation Lecture seule. S'il possède une autorisation Aucun accès sur la vue, la requête échoue comme si la vue n'existait pas.

Vous trouverez ci-dessous un exemple plus détaillé. Toutefois, pour afficher la liste complète, examinez le membre JDBCAccessor.main().

```
// Common connection properties
String userName    = "RTMadmin";
String password    = "manager";
String url = "jdbc:cognos:obi://localhost:2669";
// Verify the JDBC driver in the application
try {
   Class.forName("com.cognos.obi.jdbc.driver.Driver");
} catch (ClassNotFoundException e) {
    handleError("Could not find the JDBC driver class.",
e);
    return;
}
// Establish the connection to the JDBC driver
Connection connection = null;
try {
  connection = DriverManager.getConnection(url, userName,
password);
} catch (SQLException e) {
    handleError("Could not connect to the JDBC driver.",
e);
    return;
}
```

Exemple : Interrogation du contenu d'une vue

Cet extrait montre comment interroger l'intégralité du contenu d'une vue.

L'appel executeQuery() transmet la requête au pilote, qui renvoie le contenu de la vue dans un objet ResultSet. Toutes les colonnes du résultat de la requête sont incluses dans cette liste. Si vous définissez implicitColumns sur true, trois colonnes supplémentaires sont ajoutées. Pour plus d'informations, voir «URL JDBC (Java Database Connectivity)», à la page 243. Par ailleurs, les métadonnées sur la vue sont récupérées dans un objet ResultSetMetaData pour déterminer le nombre de colonnes de la vue.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.PrintViewContents().

```
/* Query all contents of a view.
 * Connection has already been established, and the view
name defined.
 */
String queryString = "SELECT * FROM " + VIEW_NAME ;
ResultSet rs;              // Table to hold the query
results.
```

```
ResultSetMetaData rmd;     // Metadata about the result
set.
// Query the view, and get its data and metadata.
Statement stmt = connection.createStatement();
rs = stmt.executeQuery( queryString );
rmd = rs.getMetaData();
// Print the contents of the entire view, row by row.
int columnCount = rmd.getColumnCount();
boolean isEmpty = true;
while (rs.next())  {
  isEmpty = false;
  System.out.print(" Row: ");
  for (int i=0;i<columnCount;i++) {
    // Show the column value, or "NULL"
    String ts = rs.getString(i+1);
    System.out.print( (rs.wasNull() ? "NULL " : ts +
" "));
  }
  System.out.println();  // Line break
}
if (isEmpty) {
  System.out.println("\n *** The view is empty ***");
}
```

Le résultat peut ressembler à l'exemple suivant si la valeur implicitColumns est définie sur false :

```
Contents of view [OrderProductTotals]:
 Row: Hardware Lag bolts 16400.00 41 65000.00 30000.00 100000.00
30000.00 100000.00
 Row: Hardware Nuts 337875.00 159 280000.00 240000.00 320000.00
240000.00 320000.00
 Row: Lumber Plywood 304800.00 127 250000.00 200000.00 300000.00
200000.00 300000.00
 Row: Hardware Washers 122400.00 72 170000.00 120000.00 220000.00
120000.00 220000.00
 Row: Hardware Hinges 132300.00 49 130000.00 90000.00 170000.00
90000.00 170000.00
 Row: Hardware Nails 129600.00 48 150000.00 100000.00 200000.00
100000.00 200000.00
 Row: Lumber Chip board 277200.00 126 250000.00 200000.00 300000.00
200000.00 300000.00
 Row: Hardware Screws 60000.00 30 80000.00 60000.00 100000.00 60000.00 100000.00
```

Le résultat peut ressembler à l'exemple suivant si la valeur implicitColumns est définie sur true :

```
Contents of view [OrderProductTotals]:
 Row: Hardware Lag bolts 16400.00 41 65000.00 30000.00 100000.00
30000.00 100000.00 20 2007-11-01 15:58:09.794 27
 Row: Hardware Nuts 337875.00 159 280000.00 240000.00 320000.00
240000.00 320000.00 26 2007-11-01 15:58:09.794 27
 Row: Lumber Plywood 304800.00 127 250000.00 200000.00 300000.00
200000.00 300000.00 9 2007-11-01 15:58:09.794 27
 Row: Hardware Washers 122400.00 72 170000.00 120000.00 220000.00
120000.00 220000.00 22 2007-11-01 15:58:09.794 27
 Row: Hardware Hinges 132300.00 49 130000.00 90000.00 170000.00
90000.00 170000.00 27 2007-11-01 15:58:09.794 27
 Row: Hardware Nails 129600.00 48 150000.00 100000.00 200000.00
```

```
100000.00 200000.00 25 2007-11-01 15:58:09.794 27
 Row: Lumber Chip board 277200.00 126 250000.00 200000.00 300000.00
200000.00 300000.00 17 2007-11-01 15:58:09.794 27
 Row: Hardware Screws 60000.00 30 80000.00 60000.00 100000.00 60000.00 100000.00
19 2007-11-01 15:58:09.794 27
```

Exemple : Interrogation des spécifications des colonnes d'une vue

Cet extrait montre comment interroger les spécifications définies par l'utilisateur sur les colonnes d'une vue.

Il affiche d'abord toutes les métadonnées disponibles correspondant aux spécifications des colonnes, puis les spécifications intéressantes sur chaque colonne de la vue, notamment les colonnes système internes. Les résultats apparaissent suivant l'ordre d'affichage des colonnes dans la vue, suivis des colonnes internes.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.PrintColumns().

```
/* Query a view's column specifications.
 * Connection has already been established, and the view
name defined.
 */
// Query the table to identify the columns to report
on. Because the
// view contents are irrelevant here, omit them by declaring
// 'WHERE false' as the query condition.
String queryString = "SELECT * FROM " + VIEW_NAME +" WHERE
false";
ResultSet rs;               // Table to hold the query
results.
ResultSetMetaData rmd;      // Metadata about the result
set.
// Query the view, and then get its metadata.
Statement stmt = connection.createStatement();
rs = stmt.executeQuery( queryString);
rmd = rs.getMetaData();
// Print the metadata about the columns in
the view.
System.out.println("Column details for view [" +
                 rmd.getTableName(1) +"]:");
// Walk through and show the interesting
metadata available for
// each column in the view. Include labels to identify
what we see
// in the result.
for (int i=0;i<rmd.getColumnCount();i++){
   System.out.println(
    " "             + Integer.toString(i+1) +
    ". Name ["            + rmd.getColumnName(i+1) +
    "] Type ["      + Integer.toString(rmd.getColumnType(i+1))+
    "] Precision [" + Integer.toString(rmd.getPrecision(i+1))+
    "] Scale ["     + Integer.toString(rmd.getScale(i+1))+"]");
}
```

Le résultat peut ressembler à l'exemple suivant :

```
 Column details for view [ORDERPRODUCTTOTALS]:
1. Name [FAMILY] Type [12] Precision [20] Scale [0]
2. Name [PRODUCT] Type [12] Precision [50] Scale [0]
4. Name [QTY] Type [4] Precision [10] Scale [0]
3. Name [SALES] Type [3] Precision [25] Scale [2]
6. Name [TARGET MIN] Type [3] Precision [15] Scale [2]
5. Name [TARGET] Type [3] Precision [15] Scale [2]
7. Name [TARGET MAX] Type [3] Precision [15] Scale [2]
8. Name [TARGETMIN] Type [3] Precision [15] Scale [2]
9. Name [TARGETMAX] Type [3] Precision [15] Scale [2]
```

Exemple : Interrogation des métadonnées des colonnes

Ces extraits montrent comment récupérer des métadonnées sur les colonnes d'une vue.

Pour récupérer les spécifications des colonnes, voir «Exemple : Interrogation des spécifications des colonnes d'une vue», à la page 251.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.PrintColumnMeta().

```
/* Retrieve the metadata about the columns
of a defined view.
 */
// NOTE: Specify 'null' instead of 'VIEW_NAME' to get
the metadata
//       for ALL columns in the installation.
rs = meta.getColumns(null,null,VIEW_NAME,null);
rmd = rs.getMetaData();
// Show the metadata available for view columns.
int columnCount = rmd.getColumnCount();
for (int i=0;i<columnCount;i++) {
   System.out.print(rmd.getColumnName(i+1) +
   ((i+1)==columnCount ? "" :  ","));
}
```

Les résultats répertorient d'abord les noms des colonnes des métadonnées (pour obtenir la description des colonnes, voir «Récapitulatif des colonnes de la fonction getColumns()», à la page 245) :

```
TABLE_CAT,TABLE_SCHEM,TABLE_NAME,COLUMN_NAME,DATA_TYPE,TYPE_NAME,COLUMN_SIZE,
BUFFER_LENGTH,DECIMAL_DIGITS,NUM_PREC_RADIX,NULL ABLE,REMARKS,COLUMN_DEF,
SQL_DATA_TYPE,SQL_DATETIME_SUB,CHAR_OCTET_LENGTH,ORDINAL_POSITION,IS_NULLABLE,
SCOPE_CATLOG,SCOPE_SCHEMA,SCOPE_TABLE,SOURCE_DATA_TYPE
```

Affichez ensuite l'ensemble des métadonnées sur les colonnes d'une vue particulière. Les colonnes ne s'affichent pas suivant leur ordre d'apparition dans la vue.

```
// Show the metadata values for the columns
in the view.
boolean isEmpty = true;
while (rs.next()) {
   isEmpty = false;
```

```
   for (int i=0;i<columnCount;i++) {
      String ts = rs.getString(i+1);   // Metadata value
      if (rs.wasNull())
         System.out.print(",");
      else
         System.out.print(ts +
              ((i+1)==columnCount ? "" :  ","));
   }
   System.out.println();
}
if (isEmpty) {
  System.out.println("\n *** Either there are no columns
defined "+
  "for this view (unlikely), or the view is not "+ "defined
(probably).");
}
```

Ci-après figure un exemple de liste des métadonnées de la vue OrderProductTotals. Pour obtenir une mise en correspondance des types de données Java, voir «Mappages de types de données», à la page 245. De nouveau, les colonnes ne s'affichent pas dans un ordre particulier.

```
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Family,12,VCVarchar,20,129,0,10,1,,,,,,1,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,TargetMin,3,VCDecimal,15,129,2,10,1,,,,,,2,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,TargetMax,3,VCDecimal,15,129,2,10,1,,,,,,3,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Target,3,VCDecimal,15,129,2,10,1,,,,,,4,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Sales,3,VCDecimal,25,129,2,10,1,,,,,,5,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,PRODUCT,12,VCVarchar,50,129,0,10,1,,,,,,6,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Target max,3,VCDecimal,15,129,2,10,1,,,,,,10,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Qty,4,VCInteger,10,129,0,10,1,,,,,,11,YES,,,,,
IBM Cognos RTM,IBM Cognos
RTM,OrderProductTotals,Target min,3,VCDecimal,15,129,2,10,1,,,,,,14,YES,,,,,
```

Exemple : Interrogation des métadonnées des vues

Ces extraits montrent comment interroger les métadonnées des vues.

Ils affichent d'abord les métadonnées disponibles pour les vues, puis indiquent comment rechercher toutes les vues définies dans le système.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.PrintAllViewsMeta().

```
/* Retrieve metadata about views.
 */
ResultSet rs;
ResultSetMetaData rmd;
```

```
// Uses 'null' for the 3rd parameter to retrieve information
about
//the entire system, not just a view.
DatabaseMetaData meta = connection.getMetaData();
meta = meta.getConnection().getMetaData();
rs = meta.getTables(null,null,null,new String[]{"VIEWS"});
rmd = rs.getMetaData();
for (int i=0;i<rmd.getColumnCount();i++) {
   System.out.println(" ["+ rmd.getColumnName(i+1) +
      "] Type ["       + Integer.toString(rmd.getColumnType(i+1))+
      "] Precision ["  + Integer.toString(rmd.getPrecision(i+1))
+ }
      "] Scale ["      + Integer.toString(rmd.getScale(i+1))+"]");
}
```

Les résultats décrivent d'abord les métadonnées disponibles :

```
[TABLE_CAT] Type [12] Precision [255] Scale [0]
 [TABLE_SCHEM] Type [12] Precision [255] Scale [0]
 [TABLE_NAME] Type [12] Precision [255] Scale [0]
 [TABLE_TYPE] Type [12] Precision [255] Scale [0]
 [REMARKS] Type [12] Precision [255] Scale [0]
```

Les résultats affichent ensuite les vues (noms des tables) définies dans le système :

```
// Use the metadata to list all of the views in the system.
final int GT_TABLE_NAME  = 3;  // View name
final int GT_TABLE_TYPE  = 4;  // 'VIEWS' or 'CUBES'
boolean isEmpty = true;
while (rs.next()) {
   isEmpty = false;
   System.out.println(" " + rs.getString(GT_TABLE_NAME)
+ ", " +
     rs.getString(GT_TABLE_TYPE));}
     if (isEmpty) {
        System.out.println(" *** No views are defined.
***");
    }
}
```

Le résultat peut ressembler à l'exemple suivant :

```
"30DAYORDERS", VIEWS
"8WEEKORDERS", VIEWS
"8WEEKORDERSAVG", VIEWS
INVENTORYCHANGEDETAILS, VIEWS
ORDERCHANGEDETAILS, VIEWS
ORDERPRODUCTTOTALS, VIEWS
ORDERQTYDEMAND_DRILLBACK_VIEW, VIEWS
ORDERSALESGRANDTOTAL, VIEWS
ORDERTOTALS, VIEWS
SUPPLIERALTERNATES, VIEWS
VC_SYSTEM_VIEW, VIEWS
VC_TASK_VIEW, VIEWS
```

Interfaces de cube JDBC (Java Database Connectivity)

L'interface de programmation d'applications (API) destinée aux cubes repose sur le brouillon de la future spécification olap4j version 0.5.

Le développement de la spécification olap4j n'était pas entièrement achevé au moment de l'implémentation de cette interface et cette dernière n'est pas totalement compatible avec la spécification olap4j.

Les sections ci-dessous décrivent l'API permettant d'accéder aux cubes. Pour connaître l'API complète, reportez-vous à la documentation HTML Javadoc HTML relative au pack `com.cognos.obi.jdbc.api`.

IOlapStatement

L'interface `IOlapStatement` utilise des membres de l'interface pour exécuter les requêtes.

Ces membres sont les suivants :

- `executeOlapQuery`

  Exécute une requête MDX.
- `executeOlapQueries`

  Exécute des requêtes MDX. Les ensembles de résultats renvoyés sont cohérents par rapport aux événements. Autrement dit, ils reflètent une image instantanée jusqu'au même événement.
- `executeCubeQuery`

  Exécute la requête de cube au format XML propriétaire du serveur IBM Cognos Real-time Monitoring.

IOlapDatabaseMetadata

Cette interface fournit des informations sur les définitions de cube spécifiées au moment de l'installation d' IBM Cognos Real-time Monitoring.

L'interface `IOlatDatabaseMetadata` comporte les membres suivants :

- `getCubes`

  Extrait la description d'un cube.
- `getDimensions`

  Extrait un ensemble de résultats décrivant les dimensions d'un cube particulier.
- `getHierarchies`

  Extrait un ensemble de résultats décrivant chaque hiérarchie d'une dimension particulière.
- `getLevels`

  Extrait un ensemble de résultats décrivant chaque niveau d'une hiérarchie particulière.
- `getMembers`

  Extrait un ensemble de résultats décrivant les membres d'un niveau particulier.
- `getMemberChildren`

  Extrait un ensemble de résultats décrivant les enfants d'un membre.
- `getMeasures`

  Extrait un ensemble de résultats décrivant chaque mesure d'un cube particulier.

Exemple : Interrogation de tous les cubes

Ces extraits montrent comment interroger les métadonnées des cubes.

Ils affichent d'abord les métadonnées disponibles pour les cubes, puis indiquent comment rechercher tous les cubes définis dans le système.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.PrintAllCubesMeta().

```
IOlapDatabaseMetaData meta = (IOlapDatabaseMetaData) connection.getMetaData();
rs = meta.getCubes(null, null, null);
rmd = rs.getMetaData();
for (int i=0;i<rmd.getColumnCount();i++) {
   System.out.println(
   " ["+ rmd.getColumnName(i+1) +
   "] Type ["       + Integer.toString(rmd.getColumnType(i+1))+
   "] Precision ["  + Integer.toString(rmd.getPrecision(i+1))+
   "] Scale ["      + Integer.toString(rmd.getScale(i+1))+"]");
}
```

Les résultats décrivent d'abord les métadonnées disponibles :

```
 TABLE_CAT] Type [12] Precision [255] Scale [0]
 [TABLE_SCHEM] Type [12] Precision [255] Scale [0]
 [TABLE_NAME] Type [12] Precision [255] Scale [0]
 [TABLE_TYPE] Type [12] Precision [255] Scale [0]
 [REMARKS] Type [12] Precision [255] Scale [0]
```

Les résultats affichent ensuite les cubes (noms de table, types de table et heure de la dernière modification) définis dans le système :

```
// Use the metadata to list all of the views in the system.
final int GT_TABLE_NAME  = 3;  // View name
final int GT_TABLE_TYPE  = 4;  // 'VIEWS' or 'CUBES'
final int GT_REMARKS     = 5;  // If type is 'CUBES',
                               // Remarks is the Last
modification
                               // timestamp of the cube.
                               //Otherwise, it is empty.
boolean isEmpty = true;
while (rs.next()) {
   isEmpty = false;
   System.out.println(" " + rs.getString(GT_TABLE_NAME)+
      ", " + rs.getString(GT_TABLE_TYPE)
    );
}
if (isEmpty) {
   System.out.println(" *** No views are defined. ***");
}
```

Le résultat peut ressembler à l'exemple suivant :

```
ORDERCUBE, CUBES, 2007-11-07 13:00:07.107
ORDERQTYDEMAND_DRILLBACK_CUBE, CUBES, 2007-11-07 13:00:11.2
```

Exemple : Interrogation des métadonnées d'un cube

Cet extrait montre comment récupérer les dimensions, les hiérarchies de dimension, les niveaux de structure, les membres de niveau et les mesures d'un cube.

```
ResultSet rs;
IOlapDatabaseMetaData meta = (IOlapDatabaseMetaData)connection.getMetaData();
rs = meta.getDimensions(CUBE_NAME);
System.out.println();
System.out.println("*" + rs.getMetaData().getColumnName(1)
+ ":");
boolean isEmpty = true;
while (rs.next()) {
   isEmpty = false;
   String ts = rs.getString(1);
   System.out.print("  " + ts);
   System.out.println();
   ResultSet hierarchies = meta.getHierarchies(rs.getString(1));
   System.out.println("    *" +
     hierarchies.getMetaData().getColumnName(1) +  ":");
    while (hierarchies.next()){
       System.out.print("      ");
       String hierarchy = hierarchies.getString(1);
       System.out.print(hierarchy);
       System.out.println();
       ResultSet levels = meta.getLevels(rs.getString(1),
          hierarchies.getString(1));
       System.out.println("        *" +
        levels.getMetaData().getColumnName(1) + ":");
       while (levels.next()){
          System.out.print("          ");
          String level = levels.getString(1);
          System.out.print(level);
          System.out.println();
          ResultSet members = meta.getMembers(rs.getString(1),
              hierarchies.getString(1),levels.getString(1));
          System.out.println("            *" +
            members.getMetaData().getColumnName(1) +
":");
          while (members.next()){
             System.out.print("              ");
             String member = members.getString(1);
             System.out.print(member);
             System.out.println();
          }
    }
  }
}
rs = meta.getMeasures(CUBE_NAME);
```

```
System.out.println("*" + rs.getMetaData().getColumnName(1)
+ ":");
while (rs.next()) {
   isEmpty = false;
   String ts = rs.getString(1);
   System.out.print("  " + ts);
   System.out.println();
}
if (isEmpty) {
   System.out.println("\n *** The cube was not found.");
}
```

Le résultat peut ressembler à l'exemple suivant :

```
*DIMENSION_NAME:
  DateTime Dimensions
    *HIERARCHY_NAME:
      DEFAULT HIERARCHY
        *LEVEL_NAME:
          All DateTime Dimensions level
            *MEMBER_NAME:
              All DateTime Dimensions
          Year
            *MEMBER_NAME:
              2003
              2004
              2005
          Quarter
            *MEMBER_NAME:
              Q1
              Q2
              Q3
              Q4
          Month
            *MEMBER_NAME:
              January
              February
              ...
              November
              December
          Week Number
            *MEMBER_NAME:
              1
              2
              ...
              52
              53
          Day
            *MEMBER_NAME:
              Monday
```

```
            Tuesday
            ...
            Saturday
            Sunday
 Location Dimensions
   *HIERARCHY_NAME:
     "DEFAULT HIERARCHY"
       *LEVEL_NAME:
         All Location Dimensions level
           *MEMBER_NAME:
             All Location Dimensions
         Region
           *MEMBER_NAME:
             All
             Central
             East
             South
             West
         State
           *MEMBER_NAME:
             California
             Florida
             ...
             Rhode Island
             Texas
         City
           *MEMBER_NAME:
             Austin
             Lodi
             ..
             Tug Hollow
             Wallaby Ranch
 Product Dimensions
   *HIERARCHY_NAME:
     DEFAULT HIERARCHY
       *LEVEL_NAME:
         All Product Dimensions
level
           *MEMBER_NAME:
             All Product Dimensions
         Family
           *MEMBER_NAME:
             Hardware
             Lumber
             Tape
         Product
           *MEMBER_NAME:
             Chip board
```

```
            Duct tape
            ...
            Studs
            Washers
*MEASURE_NAME:
  AverageOrderSales
  TotalOrderSales
```

Exemple : Récupération des membres enfant d'un niveau dans un cube

Ces extraits montrent comment récupérer des membres enfant d'un niveau dans un cube.

Dans cet exemple, le membre de niveau est spécifié dans la syntaxe MDX.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.testGetMemberChildren().

```
ResultSet rs;
IOlapDatabaseMetaData meta = (IOlapDatabaseMetaData)connection.getMetaData();
          String member = "[store_dimension].[All_store_dimension_level].[All_
store_dimension].[USA]";//a
member name specified in MDX syntax.
          rs = meta.getMemberChildren(member);
          int columnCount = rs.getMetaData().getColumnCount();
          for (int i=0;i<columnCount;i++) {
              System.out.print(rs.getMetaData().getColumnName(i+1));
              System.out.print(' ');
          }
          System.out.println();
          while (rs.next())  {
              System.out.print(" Row: ");
              for (int i=0;i<columnCount;i++) {
                // Show the column value, or "NULL"
                String ts = rs.getString(i+1);
                System.out.print( (rs.wasNull() ? "NULL
" : ts + " "));
              }
              System.out.println();  // Line break
            }
```

Le résultat peut ressembler à l'exemple suivant :

```
MEMBER_CHILDREN
 Row: Hinges
Row: Lag bolts
Row: Nails
Row: Nuts
Row: Screws
Row: Washers
```

Exemple : Interrogation des données de cube à l'aide du format de requête XML

Ces extraits montrent comment récupérer des données de cube à l'aide du format de requête XML, qui est le format propriétaire du serveur IBM Cognos Real-time Monitoring.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.testCubeQuery() .

```
ResultSet rs;  // Table to hold the query results.
Statement stmt = connection.createStatement();
String s = "<queryCube xsi:schemaLocation=\"http://obi.cognos.com/ ../xsd/
queryCube.xsd\"
xmlns=\"http://obi.cognos.com/4\" xmlns:xsi=\"http://www.w3.org/2001/
XMLSchema-instance\">"+
  "<cube xml:space=\"preserve\">OrderCube</cube>"
+
  "<measures>" +
    "<list>" +
     "<name xml:space=\"preserve\">TotalOrderSales</name>"+
    "</list>" +
    "<list>" +
     "<name xml:space=\"preserve\">AverageOrderSales</name>"+
    "</list>" +
   "</measures>" +
   "<dimensionLevels>" +
    "<selectedLevelsAndFilters>" +
      "<selectedLevels>" +
        "<list>" +
         "<dimensionName xml:space=\"preserve\">\"Location Dimensions\"</
dimensionName>"
+
         "<memberName xml:space=\"preserve\">Region</memberName>"+
         "<selected xml:space=\"preserve\">true</selected>"+
         "</list>" +
        "</selectedLevels>" +
       "</selectedLevelsAndFilters>" +
      "</dimensionLevels>" +
     "</queryCube>";
     rs = (ResultSet)((IOlapStatement)stmt).executeCubeQuery(s);
     int columnCount = rs.getMetaData().getColumnCount();
     for (int i=0;i<columnCount;i++) {
       System.out.print(rs.getMetaData().getColumnName(i+1));
       System.out.print(' ');
     }
     System.out.println();
     while (rs.next())  {
        System.out.print(" Row: ");
        for (int i=0;i<columnCount;i++) {
           // Show the column value, or "NULL"
           String ts = rs.getString(i+1);
```

```
          System.out.print( (rs.wasNull() ? "NULL ":
ts + " "));
         }
         System.out.println();  // Line break
     }
  }
```

Les résultats sont semblables à ceux-ci :

```
"LOCATION DIMENSIONS".REGION ORDERCUBE.TOTALORDERSALES ORDERCUBE.
AVERAGEORDERSALES
 Row: Central 286425.00 40917.85
 Row: East 107700.00 53850.00
 Row: West 986450.00 54802.77
```

Exemple : Récupération des données d'un cube à l'aide de requêtes MDX

Ces extraits montrent comment récupérer les données d'un cube à l'aide de requêtes MDX.

Pour consulter la liste complète des codes, voir l'exemple JDBCAccessor.testMDXQuery().

```
ResultSet rs;               // Table to hold the query
results.
Statement stmt = connection.createStatement();
String s = "SELECT Measures.[TotalOrderSales] ON ROWS,
[Location Dimensions].[Region].[West] ON COLUMNS FROM [OrderCube]
where [Product Dimensions].[Family].[Hardware]";
String s2 = "SELECT Measures.[TotalOrderSales] ON ROWS,
[Location Dimensions].[Region].[West].children ON COLUMNS FROM [OrderCube]
where [Product Dimensions].[Family].[Hardware]";
IOlapResultSet[] rss = ((IOlapStatement)stmt).executeOlapQueries(new
String[]{s,
s2});
for (int j = 0; j < 2; j++)
{
   System.out.println("mdx" + j + ":");
   rs = rss[j];
   int columnCount = rs.getMetaData().getColumnCount();
   for (int i=0;i<columnCount;i++) {
      System.out.print(rs.getMetaData().getColumnName(i+1));
       System.out.print(" ");
   }
   System.out.println();
   while (rs.next()){
      System.out.print(" Row: ");
      for (int i=0;i<columnCount;i++) {
         // Show the column value, or "NULL"
         String ts = rs.getString(i+1);
         System.out.print( (rs.wasNull() ? "NULL " :
ts + " "));
      }
      System.out.println();  // Line break
```

```
    }
}
```

Le résultat peut ressembler à l'exemple suivant :

```
mdx0:
"LOCATION DIMENSIONS".REGION ORDERCUBE.TOTALORDERSALES
 Row: West 610650.00
mdx1:
"LOCATION DIMENSIONS".REGION "LOCATION DIMENSIONS".STATE ORDERCUBE.TOTALORDERSALES
 Row: West California 428750.00
 Row: West Nevada 181900.00
```

Chapitre 20. Tables de correspondance

Les tables de correspondance permettent d'améliorer le traitement des informations provenant de sources d'informations grâce aux informations supplémentaires qu'elles apportent sur un événement.

Les tables de correspondance comportent des informations commerciales stockées dans les bases de données ou les entrepôts de données, ainsi que celles fournies par les services Web. Pour plus d'informations, voir Chapitre 37, «Services Web», à la page 377.Les tables de correspondance reçoivent des données provenant d'agent qui communiquent avec les sources d'informations. Pour plus d'informations, voir Chapitre 4, «Agents», à la page 25.Lorsque vous définissez une table de correspondance, vous indiquez également à l'agent les modalités d'identification des informations provenant de la source.

Fonctionnement des tables de correspondance

Lorsqu'une vue métier requiert des informations sur une table de correspondance, elle base sa recherche sur certaines des informations qu'elle contient déjà.

Par exemple, la vue qui traite un bon de commande peut recevoir un code produit avec les données d'événements. Si la vue requiert également les fournisseurs du produit, elle récupère les noms de fournisseur dans une table de correspondance contenant les noms correspondant au code produit. Dans la définition de cette vue, une clause WHERE lie la table de correspondance à l'événement, comme ci-après :

```
WHERE event.product_id = lookuptable.suppliers_of_product_id
```

Lorsque la vue effectue cette liaison, elle transmet le code produit provenant de l'événement à la table de correspondance. Si les données du fournisseur correspondant se trouvent déjà dans la mémoire cache de la table de correspondance, la table utilise ces données et les transmet à la vue métier. Si les données ne figurent pas dans la mémoire, l'ID est transmis à un agent, en tant que requête SQL ou valeur de procédure stockée, qui envoie les données au SGBD ou service Web pour traitement. Le résultat de la requête est ensuite chargé dans la table de correspondance, puis inclus dans la vue métier.

Figure 23. Fonctionnement des tables de correspondance

© Copyright IBM Corp. 2007, 2014

La source d'information peut correspondre à des bases de données accessibles par l'intermédiaire d'un JDBC (voir Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227) ou à des applications métier accessibles par le biais de services Web (voir Chapitre 37, «Services Web», à la page 377). Une source d'information peut également être un fichier texte. Pour en savoir davantage sur ces types de sources, voir «Tables JDBC (Java Database Connectivity)», à la page 227, «Tables de correspondance pour les services Web», à la page 381 ou Chapitre 13, «Fichiers à plat», à la page 103.

Création de tables de correspondance

Chaque table de correspondance comporte un attribut de nom, de description et d'état, ainsi qu'un agent.

Conseil :

Sélectionnez les index de colonnes de jointure pour les tables de correspondance utilisées par les vues ou les dimensions. Si vous n'indiquez pas d'index, la recherche de la valeur à joindre appropriée sera effectuée dans l'ensemble des lignes de la table de correspondance. L'indexation permet de détecter plus rapidement la ligne contenant la valeur appropriée.

Utilisez la mise en cache prefetch pour les tables de correspondance JDBC de sorte que le serveur n'ait pas besoin de se connecter à la base de données lorsqu'un événement se produit. A la place, les données sont examinées en bloc à des intervalles réguliers. Toutefois, n'oubliez pas que l'utilisation de la mise en cache prefetch risque d'augmenter l'utilisation de la mémoire.

Ces éléments sont décrits dans le tableau suivant :

Tableau 76. Attributs de la table de correspondance

Attribut	Description
Nom	Identifie la table et correspond au nom accessible aux vues métier qui dépendent de cette table (voir Chapitre 36, «Vues», à la page 363). Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Description	(Facultative) Description de la table.
Etat	Indique si l'objet est activé (réception et transmission de données) ou désactivé (pas de réception ni de transmission de données).
Enregistrer dans	Dossier dans lequel enregistrer la table de correspondance. Le dossier par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.

Tableau 76. Attributs de la table de correspondance (suite)

Attribut	Description
Agent	Agent chargé de récupérer les informations de correspondance et de transmettre ces dernières à l'objet de table de flux de données ou de correspondance. Pour plus d'informations sur les types d'agents, voir Chapitre 4, «Agents», à la page 25.
Désactiver la table de correspondance lorsque ce nombre d'erreurs consécutives	Nombre d'erreurs consécutives reçues du système avant que celui-ci ne désactive la table de correspondance. Une fois désactivée, une table de correspondance doit être réactivée manuellement. La valeur par défaut est 5.

Pour plus d'informations sur les types de source, voir l'exemple suivant :

- Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227
- Chapitre 37, «Services Web», à la page 377
- Chapitre 29, «Salesforce», à la page 327
- Chapitre 12, «Système de fichiers Cognos Real-time Monitoring», à la page 97

Avant de commencer

Pour créer une table de correspondance, vous devez disposer d'une autorisation de création sur les tables (voir Chapitre 23, «Autorisations», à la page 295 et «Autorisation de création d'objets», à la page 299) et d'un accès en lecture seule à l'agent qui alimente la table.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Table de correspondance**.
4. Sélectionnez le type de source.
5. Renseignez les zones de la fenêtre Nouvelle table de correspondance.
6. Enregistrez la table avec l'état activé.

Modification des tables de correspondance

La modification des attributs d'une table de correspondance entraîne la perte de l'état de l'objet et invalide éventuellement les vues dépendantes.

Par exemple, si vous supprimez une colonne, toutes les vues ou règles faisant référence à cette colonne deviennent incorrectes. (Toutefois, si vous redéfinissez la colonne dans la table, les vues dépendantes redeviennent automatiquement valides.)

Pour modifier une table de correspondance, vous devez disposer des Chapitre 23, «Autorisations», à la page 295 en lecture et en écriture pour les tables (voir «Droits d'accès», à la page 295), et d'un accès en lecture seule à l'agent qui alimente la table.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Sélectionnez la table de correspondance dans la sous-fenêtre **Objets du plan de travail** ou dans la sous-fenêtre des détails.
3. Si vous avez sélectionné la table de correspondance dans la sous-fenêtre des détails, cliquez sur **Activités**, puis choisissez **Modifier** dans le menu déroulant. Si vous avez sélectionné la table de correspondance dans la sous-fenêtre **Objets du plan de travail**, cliquez sur le bouton Modifier.
4. Modifiez les définitions dans le formulaire **Configurer la table de correspondance**. Notez que chacun des types comporte des attributs qui lui sont propres. Pour plus d'informations, consultez les sections suivantes :
   - Chapitre 18, «Connectivité JDBC (Java Database Connectivity)», à la page 227
   - Chapitre 37, «Services Web», à la page 377
   - Chapitre 13, «Fichiers à plat», à la page 103
5. Enregistrez la table avec l'état activé.

Limitations de la colonne dans les requêtes de tables de correspondance

Les données d'une table de correspondance peuvent être récupérées sans limitations à partir de la source d'une requête JDBC. Cependant, les limitations ci-dessous s'appliquent lors d'une récupération à partir d'une source de procédure stockée JDBC ou du service Web.

Les limitations sont les suivantes :

- Lorsque la colonne de la table de correspondance est référencée dans une requête, elle doit s'afficher, quelque part dans la clause WHERE ou la clause FROM, dans une expression d'égalité et uniquement comme prédicat atomique (aucun autre opérateur du même côté du signe égal). Par exemple, l'expression ci-après est autorisée :

  ```
  WHERE lookup_column =
  10*event_column
  ```

  Par contre, l'expression ci-dessous est interdite, car le prédicat du côté gauche, qui contient la colonne de la table de correspondance, est une expression comprenant un opérateur (/) :

  ```
  WHERE lookup_column/10 = event_column
  ```

- La référence atomique requise ne peut pas s'afficher dans une expression disjointe (OR). L'expression suivante échoue :

  ```
  WHERE (lookup_column
  = event_column OR A > B)
  ```

  Cependant, elle peut s'afficher dans une expression conjointe (AND) :

  ```
  WHERE (lookup_column = event_column AND A > B)
  ```

- Si la requête contient au moins une référence d'égalité, vous pouvez utiliser la colonne d'une autre manière. Par exemple, les deux requêtes ci-après sont autorisées :

  ```
  WHERE (lookup_column =
  event_column AND
         lookup_column/10 = other_event_column)
  WHERE ((lookup_column = event_column AND
  ```

```
lookup_column >= other_event_column) OR
(A > B))
```

Par contre, l'expression ci-dessous échoue car la requête ne contient pas de référence d'égalité :

```
WHERE lookup_column >= event_column
```

- L'expression d'égalité requise ne peut pas référencer une autre colonne du contexte dans la même table. Par exemple :

```
t1.lookup_column
= t1.other_lookup_column
```

- Cependant, l'expression d'égalité peut référencer une colonne du contexte dans une autre table, par exemple :

```
t1.lookup_column
= t2.other_lookup_column
```

Limitations Sybase

Lorsque vous envoyez une requête à une base de données Sybase, tenez compte des limitations suivantes :

- Tous les noms, tables et colonnes incluses, sont sensibles à la casse.
- Toutes les requêtes doivent se présenter uniquement sous la forme table SELECT * FROM. Vous ne pouvez inclure aucune clause SELECT. Pour filtrer les résultats, chargez-les dans une vue métier, puis filtrez cette vue.

Mise en cache des requêtes de tables de correspondance

La mise en cache vous permet de stocker en mémoire les résultats des requêtes de tables de correspondance.

Les recherches concernant des informations déjà récupérées sont effectuées dans les résultats stockés en mémoire plutôt que dans le SGBD à l'aide d'une nouvelle requête. Lorsque la mise en cache est active et qu'une vue requiert des données d'une table de correspondance, elle recherche d'abord dans le cache. Si le cache ne contient pas les données souhaitées, IBM Cognos Real-time Monitoring transmet une requête à la base de données si la mise en cache sur demande est activée. Toutefois, si la mise en cache prefetch est activée, Real-time Monitoring envoie une requête au cache prefetch.

Pour plus d'informations sur les méthodes de mise en cache, voir «Mise en cache sur demande», à la page 270 et «Mise en cache prefetch», à la page 270.

Les paramètres de la table de correspondance sont les suivants :

- **Mettre les données de cette table de correspondance en cache**

  Cette option détermine la mise en cache ou non des résultats de la requête sur la table de correspondance dans la mémoire cache des requêtes les plus récentes. Vous devez au préalable avoir défini ce paramètre pour pouvoir définir les autres paramètres. Vous devez également ajouter au moins un calendrier d'invalidation lorsque vous sélectionnez ce paramètre.
- **Activer prefetch**

  Cette option vous permet d'activer ou de désactiver la mise en cache prefetch des données de la table de correspondance. Lorsque vous activez cette option, la totalité de la table de correspondance externe est mise en mémoire cache à partir de la source de données externe et remplace la source des requêtes externe.
- **Nombre de résultats à mettre en cache**

Ce paramètre détermine le nombre de résultats d'une requête à mettre en mémoire cache. Chaque ensemble de résultats peut contenir une ou plusieurs lignes de données de table de correspondance liées à l'événement.

- **Calendrier d'invalidation**

  Identifie le moment opportun pour invalider le cache et supprime toutes les informations contenues dans le cache. Par exemple, tous les lundis matin à 6 heures.

Dès que vous sélectionnez une option de mise en cache, la mise en cache sur demande entre en vigueur. Par exemple, si vous sélectionnez **En mémoire** ou **Prefetch**, vous disposez de la mise en cache sur demande ainsi que de la mise en cache prefetch.

Mise en cache sur demande

La mise en cache sur demande crée un cache des requêtes les plus récentes qui gère les résultats sur une base LRU (Least Recently Used, fréquence d'utilisation moindre).

Ce cache enregistre la date à laquelle chaque résultat a été demandé. Une fois plein, le cache conserve les lignes demandées le plus récemment et efface les lignes les plus anciennes. Si les données requises ne se trouvent pas dans le cache des requêtes les plus récentes, elles sont récupérées à partir de la source de données externe.

La mise en cache sur demande se produit lorsque vous avez sélectionné l'option **Mettre les données de cette table de correspondance en cache** sur l'onglet **Mise en cache des données** et que vous ne sélectionnez pas **Activer prefetch** pour les sources JDBC ou de service Web.

Les lignes contenant des données souvent requises sont conservées le plus longtemps dans la mémoire cache des requêtes les plus récentes pour réduire l'impact sur la base de données. Toutefois, si les informations changent régulièrement, définissez un calendrier d'invalidation pour prendre en compte les modifications et invalider le cache.

Lorsqu'une invalidation planifiée se produit, le cache des requêtes les plus récentes est effacé et mis à jour avec les requêtes ultérieures. Si les données des tables de correspondance changent rarement, il est préférable d'invalider moins souvent le cache. Par exemple, si la table de correspondance est relativement statique, vous pouvez invalider le cache une fois par semaine ou par mois. Si au contraire la base de données des tables de correspondance est mise à jour tous les soirs, vous pouvez invalider le cache des requêtes les plus récentes quotidiennement pour prendre en compte les toutes dernières données.

Vous obtiendrez de meilleures performances avec la mise en cache sur demande qu'en l'absence de mise en cache. Toutefois, vous pouvez encore améliorer les performances en utilisant la mise en cache prefetch.

Mise en cache prefetch

Lorsque vous activez le cache prefetch, la totalité de la table de correspondance externe est mise en mémoire cache à partir de la source de données externe et remplace la source des requêtes externe (voir aussi Mise en cache sur demande).

Toutefois, si les données requises ne se trouvent pas dans la mémoire cache des requêtes les plus récentes, elles sont cette fois récupérées à partir du cache prefetch, et non à partir de la source de données externe.

Lorsqu'une invalidation se produit conformément au calendrier d'invalidation que vous avez défini, la mémoire cache sur demande est effacée et une requête est transmise à la source de données externe pour mettre à jour la copie en mémoire de la table de correspondance. La mémoire cache sur demande est mise à jour avec les requêtes ultérieures à partir du cache prefetch.

Pour plus d'informations sur les fonctions de cache sur demande, voir «Mise en cache sur demande», à la page 270.

Lorsque vous activez le cache prefetch, prenez en compte les effets suivants :

- La quantité de mémoire utilisée est plus importante, car les informations de la source de données externe sont stockées en mémoire et un index est créé pour chaque colonne.
- Le cache prefetch n'est mis à jour qu'au moment de l'invalidation planifiée. Si les données requises ne se trouvent pas dans le cache prefetch, aucune donnée n'est renvoyée. Les mises à jour de la source de données externe ne sont reflétées dans le cache prefetch qu'après l'invalidation planifiée suivante.

Si une exception survient, le cache prefetch effectue cinq tentatives d'actualisation du calendrier d'invalidation. Si l'actualisation échoue à partir de la source de données externe, la mémoire cache des requêtes les plus récentes est alors utilisée jusqu'à l'actualisation planifiée suivante.

Mise en cache dans la bibliothèque JDBC (Java Database Connectivity)

Vous avez la possibilité d'indiquer que les données du fichier à plat doivent être lues puis mises en cache par l'agent JDBC.

Le mécanisme d'extraction et de stockage a recours à deux tables pour stocker les données.

Une seule table à la fois peut être utilisée pour les requêtes. La table qui n'est pas utilisée par le système est purgée et rechargée conformément au calendrier d'invalidation. Une fois que ce fichier a été rechargé avec les données du fichier à plat, les requêtes sont redirigées vers la table nouvellement mise à jour, et l'ancienne table est purgée et mise à jour lors de la prochaine invalidation planifiée dans le calendrier. Par exemple, la table A est en cours d'utilisation pour les requêtes et l'invalidation est planifié à 12 heures. A l'heure indiquée, la table B est purgée et rechargée, puis les requêtes sont redirigées vers la table B. Lors de la prochaine invalidation planifiée, la table A est purgée et rechargée et les requêtes sont redirigées vers elle. La table B doit alors attendre la prochaine invalidation planifiée.

Indiquez que les données du fichier à plat sont lues et mises en mémoire par l'agent JDBC en sélectionnant le type de mise en cache **Bibliothèque JDBC**.

Chapitre 21. Objets dans Cognos Real-time Monitoring Workbench

Chaque objet a un nom, une description facultative et un état qui détermine s'il peut manipuler les données pour lesquelles il a été créé. Par ailleurs, chaque objet possède des propriétés qui définissent son comportement.

Etat d'un objet

Chaque objet possède l'un des quatre états qui déterminent son aptitude à fonctionner.

Les états sont décrits dans le tableau ci-dessous :

Tableau 77. Etats des objets

Icône	Description
	Activé L'objet accepte de nouvelles données et les traite. Vous ne pouvez activer que des objets valides qui ne dépendent pas d'objets désactivés. Lorsque vous activez un objet désactivé, vous avez le choix entre activer cet objet et activer cet objet et tous ceux qui en dépendent. Lorsque vous choisissez cette deuxième solution, les objets dépendants doivent être valides. Sinon, l'opération échoue et aucun objet n'est activé.
	Désactivé L'objet n'accepte pas de nouvelles données. La désactivation d'un objet n'affecte pas sa définition ni son existence, mais empêche les nouvelles données d'accéder à cet objet et à ceux qui en dépendent. Par exemple, la désactivation d'une vue entraîne la désactivation de toutes les règles qui la contrôlent, et aussi de toutes les alertes et tableaux récapitulatifs associés.
	Dépendant désactivé L'objet n'accepte pas de nouvelles données car un objet dont il dépend est désactivé. L'activation de l'objet référencé active également cet objet.
	Non valide L'objet comporte une référence qui n'est pas valide. Il peut s'agir d'une vue qui fait référence à une colonne d'une autre vue, alors que cette colonne n'existe plus dans la vue référencée. Un objet est non valide lorsqu'un objet référencé n'existe pas ou lorsqu'un de ses attributs ne correspond pas aux critères de l'élément dépendant. Par exemple, un type de données est discordant ou un nom de colonne est manquant. Un objet non valide entraîne également l'invalidation de tous les objets qui en dépendent. Ceci survient lorsque vous supprimez un objet qui contient des dépendances ou lorsque vous modifiez la définition d'un objet.

Lorsque vous affichez une liste d'objets, vous pouvez visualiser l'état valide/non valide et activé/désactivé de chaque objet. Vous pouvez sélectionner un objet dans la liste et l'activer via le menu **Activités**.

© Copyright IBM Corp. 2007, 2014

Noms d'objet

Un nom d'objet doit être un identificateur normal ou un identificateur délimité.

Un identificateur normal doit commencer par une lettre 'a'-'z' ou 'A'-'Z', et tous les caractères suivants peuvent être un trait de soulignement (_) ou un caractère à partir de 'a'-'z', 'A'-'Z' ou '0'-'9'. Un identificateur normal ne doit pas être un terme réservé.

Un identificateur délimité doit commencer et se terminer par un guillemet ("). Le corps de l'identificateur délimité ne doit pas être vide et peut contenir n'importe quel caractère SQL, notamment : les caractères des identificateurs normaux, le trait de soulignement (_), l'espace( ), le pourcentage(%), la perluète (&), l'apostrophe, la parenthèse ouvrante ((), la parenthèse fermante ()), l'astérisque (*), le signe plus (+), la virgule (,), le signe moins (-), la barre oblique (/), les deux points (:), le point-virgule (;), le signe égal (=), le point d'interrogation (?),la barre verticale (|) ou le guillemet ("") (accolé à un second guillemet).

Les noms de cubes ne doivent pas contenir de points (.).

Tous les noms doivent être uniques au sein de leur classe (notamment les objets de tableaux de bord et les tableaux de bord). Le dossier racine doit également contenir des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281. Les noms ne peuvent pas être identiques aux termes réservés. Par exemple, vous ne pouvez pas attribuer à une vue le nom de l'identificateur normal select, mais vous pouvez utiliser l'identificateur délimité "select".

Termes réservés

Tous les identificateurs commençant par VC_ sont des noms de système réservés et ne doivent pas être utilisés. De plus, tous les termes réservés dans la norme SQL-99 sont également réservés dans la norme C-SQL.

Le tableau suivant dresse la liste de tous les termes réservés dans IBM Cognos Real-time Monitoring.

Tableau 78. Termes réservés

Terme réservé
abs
absolute
acked
action
add
all
allocate
alter
and
any
are
as
asc

Tableau 78. Termes réservés (suite)

Terme réservé
assertion
at
authorization
avg
begin
between
bit
bit_length
boolean
both
by
cascade
cascaded
case
cast
catalog
ceil
char
char_length
character
character_length
check
close
coalesce
collate
collation
column
commit
concat
connect
connection
constraint
constraints
continue
convert
corresponding
count
create
cross
current

Tableau 78. Termes réservés (suite)

Terme réservé
current_date
current_time
current_timestamp
current_user
cursor
date
date_add
date_diff
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
domain
double
drop
else
end
end-exec
escape
event
except
exception
exec
execute
exists
exp
external
extract
false

Tableau 78. Termes réservés (suite)

Terme réservé
fetch
first
float
floor
for
foreign
found
from
full
get
global
go
goto
grant
greatest
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
into
is
is_raised
isolation
join
key
language
last
last_day

Tableau 78. Termes réservés (suite)

Terme réservé
leading
least
left
level
like
local
log
lower
lowered
lpad
ltrim
match
max
min
minute
mod
module
month
mov_avg
mov_count
mov_max
mov_min
mov_ntile
mov_rank
mov_ratio_to_report
mov_std_deviation
mov_sum
mov_variance
names
national
natural
nchar
next
no
not
ntile
null
nullif
numeric
octet_length

Tableau 78. Termes réservés (suite)

Terme réservé
of
on
only
open
option
or
order
outer
output
overlaps
pad
partial
position
power
precision
prepare
preserve
prev
primary
prior
prior_value
privileges
procedure
public
raised
rank
ratio_to_report
read
real
references
relative
restrict
revoke
right
rollback
round
rows
rpad
rtrim
schema

Tableau 78. Termes réservés (suite)

Terme réservé
scroll
second
section
select
session
session_user
set
sign
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
sqrt
std_deviation
substr
substring
sum
system_user
table
temporary
then
time
timestamp
timestamp_diff
timezone_hour
timezone_minute
to
to_char
to_date
trailing
transaction
translate
translation
trim
true
trunc

Tableau 78. Termes réservés (suite)

Terme réservé
union
unique
unknown
update
upper
usage
user
using
value
values
varchar
variance
varying
view
when
whenever
where
with
work
write
year
zone

Espace-noms d'objets

L'espace-noms contrôle la façon dont les noms sont attribués aux objets.

En général, les noms d'objet doivent être uniques parmi les objets du même type, dans le même conteneur. Il existe toutefois quelques exceptions à cette règle :

- Les alertes, les règles et les tableaux récapitulatifs peuvent partager le même nom dans le scénario qui les contient. Vous pouvez donc utiliser le même nom pour une alerte, une règle et un tableau récapitulatif au sein d'un scénario. De plus, chaque objet dans un scénario peut avoir le même nom qu'un objet du même type dans un autre scénario.
- Les profils doivent être uniques pour chaque utilisateur, mais plusieurs utilisateurs peuvent partager des noms de profil.
- Les vues, les flux de données, les tables de correspondance, etc. ne peuvent pas partager le même nom. Leurs noms doivent être uniques dans la classe des tables. Pour en savoir davantage, voir l'exemple suivant :
- Les agents doivent être uniques dans la classe des agents.
- Les utilisateurs et activités métier ne peuvent pas partager le même nom. Leurs noms doivent être uniques dans la classe des conteneurs. De plus, les utilisateurs

et les dossiers ne peuvent pas partager le même nom, car ils figurent dans le même espace-noms. Le schéma ci-dessous résume les contraintes portant sur l'espace-noms des objets :

```
/containers
    /Business activities         Unique among /containers
        /Scenarios
            /Alerts
            /Rules
            /Reportlets
    /Users and Roles             Unique among /containers
        /E-mail profiles         Unique within a user
        /RTD (Excel) profiles    Unique within a user
/tables
    /Data streams                Unique among /tables
    /Lookup tables               Unique among /tables
    /Consolidated data streams   Unique among /tables
    /Cube                        Unique among /tables
    /Dimensions                  Unique among /tables
    /Views                       Unique among /tables
/agents
    /Flat (text) files           Unique among /agents
    /SOAP (Web services)         Unique among /agents
    /JMS                         Unique among /agents
    /JDBC                        Unique among /agents
    /Rendezvous (TIBCO)          Unique among /agents
/External actions (processes)
/Join relationships
```

Exportation sélective d'objets

Vous pouvez exporter des objets sélectionnés. Cette méthode se révèle utile lorsque vous exportez des objets d'un environnement de développement vers un environnement de production.

Vous pouvez exporter un objet unique, un objet et ses dépendances, un objet et ses exigences, ou un objet et ses exigences et dépendances. Les dépendances d'un objet sont d'autres objets qui dépendent de l'objet sélectionné pour être valides. Les exigences d'un objet sont les objets dont l'objet sélectionné a besoin pour être valide.

Le diagramme ci-dessous présente les relations entre trois tableaux de bord, plusieurs objets de tableaux de bord, trois vues, deux cubes, une dimension, trois tables de correspondance, trois flux de données et trois agents. Dans ce diagramme, les dépendances de la vue 1 sont le tableau de bord 1, l'objet de tableau de bord 1, le cube 1 et l'objet de tableau de bord 2. Les exigences de la vue 1 sont le flux de données 1 et l'agent 1. Les flèches entre les objets indiquent le sens de la dépendance.

Figure 24. Relation entre différents objets

La présente section aborde les différents aspects de l'exportation sélective suivants :

- «Exemple : Exportation d'objets avec des dépendances»
- «Exemple : Exportation d'objets avec des exigences», à la page 285
- «Exemple : Exportation d'objets avec des exigences et des dépendances», à la page 286
- «Exportation d'objets sélectionnés», à la page 288

Pour plus d'informations sur l'exportation et l'importation des métadonnées Cognos Real-time Monitoring, voir le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Exemple : Exportation d'objets avec des dépendances

Vous pouvez exporter un objet avec ses dépendances. Dans certains cas, les objets inclus dans ce type d'exportation peuvent devenir non valides à la fin de l'importation. C'est parce que les exigences des dépendances peuvent ne pas être incluses dans l'exportation.

Le diagramme ci-dessous illustre deux scénarios d'exportation avec des dépendances.

Figure 25. Deux scénarios d'exportation avec dépendances

Scénario 1

Vous souhaitez exporter l'objet Flux de données 1 avec ses dépendances. Les objets suivants sont inclus ou exclus de l'exportation :

- Vue 1, Objet de tableau de bord 1, Cube 1 et Objet de tableau de bord 2 sont exportés.
- Dimension 1, Table de correspondance 1 et Agent 1 ne sont pas exportés.

  Ces objets représentent des exigences de l'objet Cube 1. Le système passe uniquement à travers les chemins de ces objets. En conséquence, l'objet Cube 1 n'est pas valide une fois l'importation terminée, car ses exigences ne sont pas incluses.
- Objet de tableau de bord 1 et Objet de tableau de bord 2 sont exportés.
- Tableau de bord 1 n'est pas exporté, même s'il contient l'objet Objet de tableau de bord 1.

Scénario 2

Vous souhaitez exporter l'objet Agent 2 avec ses dépendances. Les objets suivants sont inclus ou exclus de l'exportation :

- Les objets de tableau de bord figurant dans Tableau de bord 2 et Tableau de bord 3 sont exportés, mais les tableaux de bord eux-mêmes ne sont pas exportés.
- Vue 2 et Vue 3 sont exportés.

- Flux de données 2 et Agent 1, qui sont des exigences de Vue 2, ne sont pas exportés.

  En conséquence, Vue 2 n'est pas valide à l'issue de l'importation.
- Agent 3 et Table de correspondance 3, qui sont des exigences de Vue 3, ne sont pas exportés.

  En conséquence, Vue 3 n'est pas valide à l'issue de l'importation.

Exemple : Exportation d'objets avec des exigences

Vous pouvez exporter un objet avec ses exigences.

Le diagramme ci-dessous illustre quelques exemples de ce type d'exportation.

Figure 26. Exemples d'objets exportés avec des exigences

Lors de l'exportation de l'objet Vue 1 avec ses exigences, les objets Flux de données 1 et Agent 1 sont également exportés.

Lors de l'exportation de l'objet Cube 2 avec ses exigences, les objets Dimension 1, Table de correspondance 1, Agent 1, Vue 2, Flux de données 2, Table de correspondance 2 et Agent 2 sont également exportés.

Lors de l'exportation de l'objet Tableau de bord 3 avec ses exigences, les objets Objet de tableau de bord 4, Objet de tableau de bord 5 et Objet de tableau de bord

6 sont également exportés. De plus, les exigences de ces trois objets de tableau de bord, les objets Vue 3, Flux de données 3, Table de correspondance 3, Agent 2 et Agent 3, sont exportés.

Exemple : Exportation d'objets avec des exigences et des dépendances

Dans certains cas, vous pouvez exporter les dépendances et les exigences d'un objet.

Les scénarios suivants expliquent les conséquences liées à ce type d'exportation.

Scénario 1

Vous souhaitez exporter la dimension 1 avec ses dépendances et ses exigences, comme illustré dans le diagramme ci-dessous.

Les objets suivants sont inclus ou exclus de l'exportation :

- Objet de tableau de bord 2, Objet de tableau de bord 3, Cube 1 et Cube 2 sont exportés.

  Il s'agit des dépendances de la dimension 1.
- Table de correspondance 1 et Agent 1 sont exportés.

  Il s'agit des exigences de la dimension 1.
- Vue 1 et Vue 2 ne sont pas exportées.

Il s'agit des exigences liées aux objets Cube 1 et Cube 2. Lorsque vous n'incluez pas ces objets, Cube 1 et Cube 2 deviennent non valides après l'importation. Une seule exigence Cube 1 et Cube 2 (Dimension 1) est exportée.

Scénario 2

Vous souhaitez exporter la dimension 1 avec ses dépendances et les exigences de ces dépendances, comme illustré dans le diagramme ci-dessous.

Astuce : Pour réaliser ce scénario, utilisez l'option **Exporter un sous-projet**.

Les objets suivants sont inclus ou exclus de l'exportation :

- Objet de tableau de bord 2, Objet de tableau de bord 3, Cube 1 et Cube 2 sont exportés.

  Il s'agit de toutes les dépendances de la dimension 1.
- Table de correspondance 1 et Agent 1 sont exportés.

  Il s'agit des exigences de la dimension 1.
- Vue 1, Vue 2, Flux de données 1, Flux de données 2, Table de correspondance 2 et Agent 2 sont exportés.

  Il s'agit des exigences requises pour que les objets Cube 1 et Cube 2 soient valides.
- Les dépendances des objets Vue 1 et Agent 2 ne sont pas exportées.

Scénario 3

Vous souhaitez exporter la dimension 1 avec l'ensemble de ses dépendances et exigences, ainsi que les dépendances et les exigences de tous les objets associés, comme illustré dans le diagramme ci-dessous.

Astuce : Pour réaliser ce scénario, utilisez l'option **Exporter la totalité du projet**.

Lorsque vous appliquez ce scénario, tous les objets sont valides après leur importation.

La totalité des objets du Projet Un sont exportés. Les objets du Projet Deux ne sont pas exportés, car aucun objet du Projet Un n'est connecté à un objet du Projet Deux.

Exportation d'objets sélectionnés

Vous pouvez exporter des objets sélectionnés.

L'action d'exportation écrit les informations d'objet dans l'un des types de fichiers suivants :

- Les fichiers XML d'un répertoire sur le serveur.
- Un fichier JAR compressé unique sur le serveur ou sur un client local.

Pour plus d'informations sur l'exportation et l'importation des métadonnées IBM Cognos Real-time Monitoring, voir le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Avant de commencer

Avant d'exporter les objets, vérifiez que vous avez bien compris les dépendances et les exigences correspondantes.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Bibliothèque d'objets**.
3. Sélectionnez un objet.
4. Cliquez sur **Activités** > **Exporter des objets**.
5. Dans la fenêtre Exportation sélective, depuis le menu déroulant **Opérations**, sélectionnez l'un des types d'exportation suivants :
   - **Exporter les métadonnées dans un répertoire du serveur**

     Cette option permet d'exporter les métadonnées dans le répertoire que vous indiquez dans la zone **Emplacement du serveur**. Tapez le chemin d'accès complet au répertoire dans lequel vous souhaitez exporter les métadonnées. Ce répertoire doit exister et vous devez disposer des droits d'accès en écriture vers ce répertoire. De même, deux répertoires sont créés dans le répertoire que vous spécifiez. Un sous-répertoire est nommé "latest". L'autre correspond à l'horodatage de votre exportation. Si les répertoires existent, ils sont réutilisés.
   - **Exporter les métadonnées dans un fichier JAR du serveur**

     Cette option permet d'exporter les métadonnées vers un fichier JAR du répertoire que vous indiquez dans la zone **Emplacement du serveur**. Tapez le chemin d'accès complet au répertoire dans lequel vous souhaitez exporter les métadonnées. Ce répertoire doit exister et vous devez disposer des droits d'accès en écriture vers ce répertoire.

     Le nom du fichier sur le serveur est _export.jar. Si le fichier _export.jar existe, il est remplacé.
   - **Exporter les métadonnées dans un fichier JAR (télécharger)**

     Cette option permet d'exporter les métadonnées vers un fichier JAR sur le client. Utilisez la boîte de dialogue de téléchargement de votre navigateur pour indiquer l'emplacement (et éventuellement modifier le nom du fichier) sur votre ordinateur local.
6. Dans le menu déroulant **Options d'exportation**, sélectionnez l'une des options suivantes :
   - **Exporter la totalité du projet**

     Exporte les objets sélectionné, ainsi que les dépendances et exigences de tous les objets associés.
   - **Exporter un sous-projet**

     Exporte les objets sélectionnés avec leurs exigences et dépendances, ainsi que les exigences de ces dépendances.
   - **Exporter avec les exigences**

     Exporte les objets sélectionnés avec leurs exigences.
   - **Exporter avec les dépendances**

Exporte les objets sélectionnés avec leurs dépendances. Les exigences des dépendances ne sont pas exportées.

- **Exporter avec les exigences et les dépendances**

  Exporte les objets sélectionnés avec leurs dépendances et exigences. Les exigences des dépendances ne sont pas exportées.

- **Exporter uniquement les objets sélectionnés**

  Exporte uniquement les objets que vous sélectionnez. Les dépendances et exigences de ces objets ne sont pas exportées.

7. Facultatif : sélectionnez **Inclure des droits à ces objets**.

   Cette option est utile lorsque vous souhaitez éviter de réaffecter des droits sur vos objets après les avoir importés.

8. Cliquez sur **Aperçu de la liste d'exportation** afin de vous assurer que vous avez sélectionné les options d'exportation appropriées.
9. Cliquez sur **OK**.

Chapitre 22. Opérateurs et constantes

IBM Cognos Real-time Monitoring et C-SQL prennent en charge les opérateurs et constantes ci-dessous dans les expressions et les arguments.

Opérateurs numériques

Il existe deux classes d'opérateurs numériques : préfixe et infixe.

Opérateurs préfixe

Les opérateurs préfixe contrôlent le signe arithmétique des valeurs numériques :

Tableau 79. Opérateurs préfixe

Opérateur préfixe	Description
+	Plus unaire
-	Moins unaire

Opérateurs infixe

Les opérateurs numériques effectuent des opérations arithmétiques sur les valeurs numériques :

Tableau 80. Opérateurs infixe

Opérateur numérique	Description	Exemple
+	Addition	11 + 3 = 14
-	Soustraction	11 - 3 = 8
*	Multiplication	11 * 3 = 33
/	Division	11 / 3 = 3

Pour effectuer des opérations arithmétiques sur des valeurs date-heure, utilisez les fonctions DATE_ADD et DATE_DIFF. Pour en savoir davantage, reportez-vous aux sections «DATE_ADD», à la page 144 et «DATE_DIFF», à la page 145.

Opérateurs de chaîne

L'opérateur de concaténation (|| fn_||) est le seul opérateur de chaîne. Il ajoute la chaîne située à droite de l'opérateur à la fin de celle située à gauche. Par exemple, 'a'||'b' renvoie 'ab'.

Ce comportement est identique à celui de la fonction CONCAT. Pour en savoir davantage, voir «CONCAT», à la page 168.

© Copyright IBM Corp. 2007, 2014

Opérateurs de comparaison

Les opérateurs de comparaison comparent plusieurs valeurs du même type de données et renvoient une valeur booléenne.

Tableau 81. Opérateurs de comparaison

Opérateur	Description
=	Egal à Exemple : 1=1 renvoie TRUE
<>	Non égal à Exemple : 'A'<>'a' renvoie TRUE
>	Supérieur à Exemple : CURRENT_DATE()>TO_DATE('02/28/1963') renvoie TRUE
<	Inférieur à Exemple : Age<21 renvoie UNKNOWN lorsque Age a pour valeur NULL
>=	Supérieur ou égal à Exemple : TRUE>=FALSE renvoie TRUE
<=	Inférieur ou égal à Exemple : NULL<=NULL renvoie NULL
IN	Est un membre de la liste Exemple : Symbole IN ('IBM', 'MSFT', 'VCLR') ou Nombre NOT IN (5, 10, 15, 20)
BETWEEN /AND	Est dans une plage Exemple : SalePrice BETWEEN 50.0 AND (90.0) ou NOT BETWEEN 'M' AND 'O'
LIKE	Modèle de correspondance Exemple : Titre LIKE 'MR_'

Opérateur LIKE

L'opérateur LIKE recherche une suite de caractères correspondants. Dans la chaîne, un signe de pourcentage (%) est une valeur générique correspondant à zéro ou à plusieurs caractères, et un trait de soulignement (_) est une valeur générique correspondant à un seul caractère.

```
WHERE Title LIKE 'MR_'
WHERE E_Mail NOT LIKE '%.edu'
```

Pour inclure les signes pourcentage (%) ou trait de soulignement (_) dans la chaîne recherchée, utilisez le mot clé ESCAPE pour désigner un caractère d'échappement. Le signe % ou _ situé après un caractère d'échappement est considéré comme un littéral. Notez qu'il est alors impossible d'utiliser le caractère d'échappement à un autre endroit dans la chaîne recherchée. L'exemple ci-dessous recherche 10% n'importe où dans la chaîne Discount :

```
WHERE Discount LIKE '%10$%' ESCAPE '$'
```

Un caractère d'échappement placé avant un caractère autre qu'un caractère d'échappement ou un caractère spécial est ignoré.

Soyez prudent si vous utilisez l'opérateur LIKE pour effectuer des comparaisons avec des types numériques. LIKE étant un opérateur de chaîne, toute recherche d'une valeur numérique est précédée d'une conversion implicite de la valeur numérique en une chaîne. Pour en savoir davantage, voir «Conversion implicite», à la page 69. Lors de la conversion de valeurs numériques en chaînes, tenez compte des points suivants :

- La conversion d'un nombre DECIMAL en une chaîne ajoute des zéros à cette valeur décimale en fonction de la précision et de l'échelle définies pour la colonne. Par exemple, si une colonne a une précision de 5 et une échelle de 4, la valeur 1.1 dans la colonne devient 1.1000. Ainsi, pour rechercher des nombres à une décimale, vous devez formuler l'équation LIKE '_._000'.
- Pour les nombres à DOUBLE PRECISION, le 'e' est converti en majuscule. Par exemple, +1e11 est converti en 1.0E11.

Opérateurs logiques

Les opérateurs logiques comparent des valeurs booléennes, comme le résultat d'une opération de comparaison.

Tableau 82. Opérateurs logiques

Opérateur logique	Description
AND	Les deux conditions sont remplies Exemple : (SalesPrice>500) AND (OnSale)
OR	Une des deux conditions doit être remplie Exemple : (ZipCode = '90210') OR (City = 'Lodi')
NOT	Inverse Exemple : NOT OnSale
IS	Test de valeur booléenne fn_is not Exemple : IS OnSale ou IS NOT OnSale

La table de vérité de l'opérateur d'égalité (=) est équivalente à IS. Pour en savoir davantage, voir «Opérateurs booléens», à la page 82.

Constantes

Les constantes booléennes C-SQL sont TRUE, FALSE et UNKNOWN.

Pour en savoir davantage, voir «Opérateurs booléens», à la page 82.

Tableau 83. Constantes

Constantes	Description
TRUE	Condition vérifiée
FALSE	Condition non vérifiée
NULL	Aucune donnée
UNKNOWN	Test d'une valeur booléenne de type NULL ou impossible à comparer, comme dans le cas d'une comparaison de deux valeurs NULL.

La valeur NULL est ignorée lors du calcul de valeurs à l'aide de la fonction d'ensembles (Set), de la fonction d'ensembles mobiles (Moving set) et de la fonction de classement (Rank). Par exemple, la moyenne de (3, NULL, 3) est 3, pas NULL, ni 2.

NULL représente une valeur de type null. Toute colonne de type non booléen qui ne contient aucune entrée est considérée comme de type NULL. Pour en savoir davantage, voir «Opérateurs booléens», à la page 82.

```
WHERE SalesPrice = NULL     Returns TRUE
WHERE SalesPrice = UNKNOWN  Error,
cannot cast Numeric to Boolean
```

Toutefois, le test d'une colonne de type booléen renvoie UNKNOWN lorsqu'elle est vide. Pour en savoir davantage, voir «Opérateurs booléens», à la page 82.

```
WHERE OnSale = NULL         Returns UNKNOWN (null = null)
WHERE OnSale = UNKNOWN      Returns UNKNOWN (null = unknown)
WHERE OnSale IS NULL        Returns TRUE
WHERE OnSale IS UNKNOWN     Returns TRUE
```

Chapitre 23. Autorisations

Les autorisations identifient les utilisateurs qui sont habilités à voir, créer et modifier des objets et des comptes utilisateur.

Les autorisations sont appliquées à partir de plusieurs niveaux :

- A partir de l'onglet **Droits d'accès** de la fenêtre Modifier l'utilisateur. Vous accédez à cette fenêtre à partir de l'onglet **Console d'administration** en sélectionnant le dossier **Utilisateurs** et en cliquant deux fois sur un nom d'objet. L'onglet **Droits d'accès** vous permet d'afficher et d'affecter les droits de niveau type pour l'utilisateur sélectionné. L'autorisation de niveau type que vous attribuez correspond à l'autorisation minimum de l'utilisateur sur tous les objets de ce type.
- A partir de l'onglet **Droits d'accès** de la fenêtre Modifier le rôle. Vous pouvez attribuer des autorisations de niveau type à un rôle en cliquant deux fois sur le rôle. Si vous attribuez des droits à un rôle, tous les utilisateurs associés à ce rôle disposent implicitement de ces autorisations.
- A partir de la fenêtre Droits. Pour y accéder, cliquez sur **Droits** dans la vue détaillée d'un objet.

Lorsqu'un utilisateur possède un ou plusieurs rôles, c'est le niveau d'accès le plus élevé entre les rôles et les autorisations attribuées qui s'applique. Pour en savoir davantage, voir Chapitre 27, «Rôles», à la page 317. La fenêtre Droits affiche uniquement les autorisations explicitement affectées.

Par défaut, chaque nouvel utilisateur possède l'autorisation **Aucun accès** sur tous les objets (à l'exception des comptes utilisateur sur lesquels il possède l'autorisation **Lecture seule** : utilisateur, rôles et propriétés système globales). Toutefois, même avec ces autorisations, un utilisateur peut recevoir et afficher des notifications d'alerte et les tableaux récapitulatifs générés dans le cadre d'abonnements obligatoires.

L'utilisateur rtmadmin possède toutes les autorisations sur l'ensemble des objets dans l'installation. Pour plus d'informations sur l'utilisateur rtmadmin, voir Chapitre 34, «Utilisateurs», à la page 355.

Droits d'accès

Les droits d'accès déterminent le niveau d'accès d'un utilisateur sur un objet. Ils peuvent être attribués à une classe d'objets ou à un objet spécifique.

Les droits d'accès sont les suivants :

Tableau 84. Droits d'accès

Autorisation	Type d'objet	Objet
Aucun accès	L'utilisateur ne peut pas voir les objets de cette classe, sauf s'il a reçu l'autorisation Lecture pour certains objets.	Non applicable

© Copyright IBM Corp. 2007, 2014

Tableau 84. Droits d'accès (suite)

Autorisation	Type d'objet	Objet
Filtré/Lecture seule	Non applicable	Limite, dans une vue, les lignes que l'utilisateur peut consulter, en fonction d'un filtre d'accès. Pour en savoir davantage, voir Chapitre 2, «Filtres d'accès», à la page 3.
Lecture seule	L'utilisateur peut afficher tous les objets de ce type.	L'utilisateur peut voir l'objet.
Lecture et écriture	L'utilisateur peut afficher et modifier tous les objets de ce type.	L'utilisateur peut afficher et modifier l'objet.

Vous ne pouvez pas attribuer une autorisation supérieure à la vôtre sur un même objet.

Droits d'accès au niveau du type

Toutes les autorisations de niveau type sont attribuées à un utilisateur ou un rôle. Pour afficher ou modifier une autorisation de niveau type, vous devez tout d'abord modifier le compte de l'utilisateur ou la définition du rôle.

Lorsque vous consultez la liste des utilisateurs ou des rôles, ne cliquez pas sur le bouton **Droits**. Ce bouton identifie les utilisateurs et les rôles ayant accès à certains comptes utilisateur ou à des définitions de rôle dans la liste. Pour plus d'informations, voir «Droits d'accès à certains objets», à la page 297.

Modification des autorisations d'un utilisateur sur une classe d'objets

Vous pouvez modifier les autorisations d'un utilisateur sur une classe d'objets.

Pour en savoir davantage, voir «Onglet Droits d'accès», à la page 357.

Procédure

1. Dans l'onglet **Console d'administration**, cliquez sur **Utilisateurs**.
2. Cliquez deux fois sur le compte utilisateur dans la liste.
3. Dans la fenêtre Modifier l'utilisateur, cliquez sur l'onglet **Droits d'accès**.

   Cet onglet affiche les autorisations en vigueur, y compris celles affectées aux rôles auxquels l'utilisateur appartient.
4. Cliquez sur **Modifier** en regard de l'autorisation à modifier. La fenêtre des autorisations comporte trois zones :
   - La zone **Droits inhérents au rôle** affiche l'autorisation attribuée par les rôles dont l'utilisateur est détenteur. Lorsqu'un utilisateur possède plusieurs rôles, c'est le niveau d'accès le plus élevé qui est appliqué.
   - La zone **Droits supplémentaires spécifiques à l'utilisateur** représente l'autorisation que vous attribuez à l'utilisateur sur la classe. Lorsque vous attribuez une autorisation inférieure à celles du rôle, vous ne restreignez pas les droits d'accès de l'utilisateur.
   - La zone **Droits effectifs** indique le plus haut niveau d'autorisation attribué par les deux autres zones et correspond à l'autorisation attribuée à l'utilisateur sur cette classe d'objets.
5. Enregistrez les modifications pour les appliquer immédiatement à l'utilisateur.

Modification des autorisations d'un rôle sur une classe d'objets

Vous pouvez modifier les autorisations d'un rôle sur une classe d'objets.

Procédure

1. Dans l'onglet **Console d'administration**, cliquez sur **Rôles**.
2. Cliquez deux fois sur le compte utilisateur dans la liste.
3. Dans l'onglet **Droits d'accès**, cliquez sur **Modifier** en regard de l'autorisation à modifier. Définissez l'autorisation de ce rôle sur la classe.
4. Enregistrez les modifications.

Modification des autorisations d'un utilisateur pour la création de rapports

Vous pouvez modifier les autorisations d'un utilisateur pour la création de rapports.

Procédure

1. Dans l'onglet **Console d'administration**, cliquez sur **Utilisateurs**.
2. Cliquez deux fois sur le compte utilisateur dans la liste.
3. Dans la fenêtre Modifier l'utilisateur, cliquez sur l'onglet **Droits d'accès**.
4. Cliquez sur **Modifier** en regard de l'option Créer des rapports. La fenêtre Autoriser à créer des rapports s'affiche. Elle vous permet de définir les autorisations de création d'un **rapport** et d'un **rapport avancé**. Grâce au droit d'accès, l'utilisateur peut créer un rapport. Grâce au droit d'autorisation, l'utilisateur peut créer un rapport et accorder une autorisation de création de rapport à un autre utilisateur.
5. Cliquez sur **OK**.

Droits d'accès à certains objets

Pour accéder aux autorisations définies sur certains objets, sélectionnez un objet dans la liste et cliquez sur le bouton Droits.

Notez que cette instruction s'applique à la fois aux listes Utilisateurs et Rôles. La procédure d'attribution de droits d'accès à certains comptes utilisateur et rôles est identique à celle qui permet de définir les droits d'accès à des vues ou des agents.

Affichage des autorisations d'un utilisateur sur un ou plusieurs objets spécifiques

Vous pouvez afficher les autorisations d'un utilisateur sur un ou plusieurs objets spécifiques.

Procédure

1. Sélectionnez les objets concernés dans la liste, notamment une ou plusieurs vues spécifiques.
2. Cliquez sur **Activités**.
3. Sélectionnez **Droits**. La fenêtre correspondante fournit les informations suivantes :
   - **Vos droits**

     Droits d'accès effectifs, que vous possédez sur cet objet, en tant qu'utilisateur connecté à IBM Cognos Real-time Monitoring Workbench. Les autorisations effectives représentent une combinaison des autorisations qui vous ont été attribuées explicitement ou implicitement via les rôles auxquels vous appartenez sur cet objet ou ce type d'objet.

- **Nom**

  Nom des utilisateurs et des rôles qui possèdent une autorisation sur cet objet ou ce type d'objet. Dans le cadre des utilisateurs, seuls ceux disposant d'autorisations explicites sont répertoriés.
- **Type**

  Type indiquant si la valeur **Nom** est associée à une valeur **Rôle** ou **Utilisateur**.
- **Droits d'accès**

  Droits d'accès explicitement attribués au **Rôle** ou à l'**Utilisateur** sur cet objet. Cliquez sur le bouton **Modifier les droits** pour définir les droits d'un **Utilisateur** ou d'un **Rôle**.
- **Droits d'accès au niveau du type**

  Droits d'accès de ce type d'objet attribués de façon explicite à l'**Utilisateur** ou au **Rôle**.

Modification des autorisations d'un utilisateur sur un ou plusieurs objets spécifiques

Vous pouvez modifier les autorisations d'un utilisateur sur un ou plusieurs objets spécifiques.

Procédure

1. Sélectionnez un ou plusieurs objets et choisissez **Droits** dans la liste déroulante **Activités**. L'autorisation **Filtré / Lecture seule** ne peut être attribuée qu'à une vue ou à un cube à la fois.

   Dans le cas d'un objet unique, vous pouvez aussi cliquer sur le bouton **Droits** de la vue détaillée de l'objet.
2. Sélectionnez les utilisateurs dont vous souhaitez modifier les droits d'accès et cliquez sur **Modifier les droits**.

   Vous n'êtes pas autorisé à modifier les autorisations des utilisateurs qui ne répondent pas aux critères spécifiés dans la section «Restrictions d'autorisation» , à la page 300. Vous ne pouvez pas modifier les autorisations des utilisateurs qui possèdent le symbole d'interdiction en regard de leur nom.
3. Définissez les autorisations dans l'onglet **Simple**. Notez que lorsque vous attribuez des autorisations sur une vue ou un cube, vous avez la possibilité de choisir l'autorisation **Filtré / Lecture seule**. Pour plus d'informations sur cette autorisation, voir Chapitre 2, «Filtres d'accès», à la page 3.

   Enregistrez les autorisations pour les appliquer immédiatement aux objets.

Ajout d'un utilisateur ou d'un rôle aux autorisations d'un ou de plusieurs objets

Vous pouvez ajouter un utilisateur ou un rôle aux autorisations liées à un ou plusieurs objets.

Procédure

1. Sélectionnez un ou plusieurs objets et choisissez **Droits** dans la liste déroulante **Activités**.

   Dans le cas d'un objet unique, vous pouvez cliquer sur le bouton **Droits** de la vue détaillée de l'objet.
2. Dans la boîte de dialogue Droits, cliquez sur le bouton **Ajouter un utilisateur** ou **Ajouter un rôle** en fonction de l'action que vous souhaitez exécuter.
3. Cochez la case en regard du nom à ajouter aux permissions.

Autorisation de création d'objets

Les autorisations de création déterminent les classes d'objets qu'un utilisateur est autorisé à créer.

Lorsque vous créez un objet, vous possédez les autorisations de Lecture et écriture et les droits d'autorisation sur cet objet. Ceci vous permet d'accorder les droits d'accès et d'autorisation à tous les autres utilisateurs de cet objet.

Lorsque vous avez créé un objet, tout autre utilisateur possédant les droits d'autorisation sur l'objet peut réattribuer des autorisations et modifier celles que vous avez définies.

Lorsque vous créez un utilisateur, celui-ci possède l'autorisation Aucun accès sur tous les objets. Vous pouvez attribuer à cet utilisateur des autorisations à une classe complète d'objets, grâce à l'onglet relatif aux droits d'accès de l'utilisateur. Pour en savoir davantage, voir «Onglet Droits d'accès», à la page 357.

Affichage des types d'objets que vous êtes autorisé à créer

Vous pouvez afficher les types d'objets que vous êtes autorisé à créer.

Procédure

Cliquez sur **Paramètres du compte** et affichez l'onglet Droits d'accès de votre compte.
Les autorisations effectives que vous obtenez explicitement ou implicitement via les rôles auxquels vous appartenez s'affichent. Pour en savoir davantage, voir «Onglet Droits d'accès», à la page 357.

Droits d'autorisation

Les droits d'autorisation vous permettent d'attribuer des autorisations à d'autres utilisateurs.

Ces droits sont les suivants :

Tableau 85. Droits d'autorisation

Autorisation	Description
Aucune capacité à accorder	Ne permet pas d'accorder des autorisations.
Capacité à accorder des droits d'accès en lecture seule	Permet d'attribuer l'autorisation Lecture seule.
Capacité à accorder des droits d'accès en lecture	Permet d'autoriser l'attribution Lecture seule.
Capacité à accorder des droits d'accès en lecture et en écriture	Permet d'attribuer l'autorisation Lecture et écriture.
Capacité à accorder des droits d'accès en lecture et en écriture	Permet d'autoriser l'attribution Lecture et écriture.

L'utilisateur rtmadmin est le seul utilisateur qui possède toutes les autorisations sur tous les objets.

Procédure

1. Sélectionnez les objets et cliquez sur **Droits**.

2. Sélectionnez les utilisateurs à modifier, cliquez sur **Modifier les droits** et choisissez les droits d'autorisation dans l'onglet **Avancé**.

Restrictions d'autorisation

Lorsque vous octroyez des droits d'autorisation, gardez à l'esprit que certaines restrictions s'appliquent.

Ces restrictions sont les suivantes :

- Vous ne pouvez pas réduire l'autorisation d'un autre utilisateur sur un objet pour lequel il possède des droits d'accès supérieurs aux vôtres.
- Vous ne pouvez pas attribuer à un autre utilisateur des droits d'accès supérieurs aux vôtres sur un objet donné. Cette restriction s'applique lorsque vous tentez d'attribuer des droits d'accès à plusieurs objets à la fois, alors que vos droits d'accès à l'un de ces objets sont inférieurs aux vôtres pour les autres objets.

Héritage et dépendances des autorisations

Les objets qui gèrent les autorisations contrôlent l'accès à l'objet, ainsi qu'aux objets qu'il contient. D'autres objets héritent leurs autorisations de l'objet qui les contient.

Les objets qui contrôlent les autorisations sont les suivants :

- Agents
- Activités métier (qui contrôlent l'accès aux scénarios, règles, alertes et tableaux récapitulatifs contenus)
- Rôles
- Tables, y compris les tables de correspondance, de flux de données et de flux de données consolidées
- Utilisateurs (contrôlent l'accès aux comptes utilisateur)
- Vues, cubes et dimensions
- Tableaux de bord
- Objets de tableau de bord
- Processus externes
- Rapports de création

Quelles que soient ses autorisations sur une alerte ou un tableau récapitulatif, un utilisateur peut toujours consulter les informations fournies dans les notifications d'alerte et les tableaux récapitulatifs qui lui sont envoyés.

Dépendances

Lorsque vous créez ou modifiez des objets, vous êtes limité par les autorisations des objets dépendants et celles de l'objet contenant.

Les objets dont les autorisations de création ou de modification présentent des dépendances sont les suivants :

Table Vous devez posséder l'autorisation Lecture sur l'agent qui alimente la table.

Vue Vous devez posséder l'autorisation **Lecture** sur la table de flux de données ou la vue de base et sur les tables de correspondance qui alimentent la vue.

Cubes Vous devez posséder l'autorisation **Lecture** sur la table de flux de données ou la vue de base et sur les dimensions qui alimentent la vue.

Activité métier

Pour afficher les définitions des objets contenus dans une activité métier, vous devez posséder l'autorisation **Lecture** sur cette activité métier. Pour afficher les définitions des scénarios, règles, alertes et tableaux récapitulatifs contenus, vous devez posséder l'autorisation **Lecture** sur la vue sous-jacente. Pour créer un objet contenu, vous devez posséder l'autorisation **Lecture et écriture** sur l'activité métier, ainsi que l'autorisation **Lecture** sur la vue sous-jacente.

Les autorisations sur un objet peuvent modifier l'accès d'un utilisateur à un autre objet, notamment pour les restrictions concernant les vues. Par exemple, vous pouvez être autorisé à modifier une alerte mais pas son tableau récapitulatif, lorsque l'autorisation définie pour la vue de ce dernier est **Aucun accès**, mais que l'autorisation définie pour la vue de l'alerte est **Lecture seule**.

Chapitre 24. Intégration du serveur de portail

Vous pouvez intégrer des serveurs IBM Cognos Real-time Monitoring dans l'un des serveurs de portail pris en charge, conformément à la spécification JSR-168.

Prérequis pour l'intégration du serveur de portail

Plusieurs conditions requises doivent être remplies avant de démarrer la procédure d'intégration du serveur de portail.

Ces conditions sont les suivantes :

- Les serveurs IBM Cognos Real-time Monitoring doivent être démarrés et en cours d'exécution sur l'une des implémentations du serveur d'applications prises en charge.
- L'un des serveurs de portail pris en charge suivants doit être installé et en cours d'exécution :

Serveur de portail	Description
Serveur de portail BEA WebLogic	Version livrée avec WebLogic Application Server v 8.1 SP5. Veillez à installer l'application WebLogic Workshop également fournie, car celle-ci sera requise pour effectuer plusieurs étapes de l'implémentation.
Serveur de portail IBM WebSphere	Version 5.1.0.1 Ce produit est livré avec WebSphere Application Server version 5.1.13. (DB2 uniquement) Si vous utilisez WebSphere Application Server avec une base de données DB2, vous devez également installer DB2 Universal V 8.1 ou 8.2 (groupe de correctifs 10).
Serveur de portail JBoss	Version 2.2.1 SP3 Ce produit est livré avec le serveur d'applications JBoss 4.0.3 SP1.

Intégration dans le serveur de portail BEA WebLogic

Cette section décrit la procédure de conversion et de déploiement du fichier `lavaJSR168.war` dans le serveur de portail WebLogic.

Conversion du fichier lavaJSR168.war

Pour procéder à l'intégration dans le serveur de portail WebLogic, vous devez tout d'abord convertir le fichier `lavaJSR168.war` dans la spécification du portail WebLogic à l'aide de l'utilitaire Portlet Preparer Tool disponible auprès de BEA.

Procédure

1. A l'aide de l'utilitaire Portlet Preparer Tool, extrayez le fichier `portletConverter.jar` dans un répertoire temporaire : $CONVERTER_ROOT$.
2. Copiez le fichier `lavaJSR168.war` dans ce même répertoire.
3. Dans l'utilitaire Portlet Preparer Tool, faites pointer WEBLOGIC_HOME dans `env.sh` ou `env.bat` (selon la plateforme) vers ce même répertoire temporaire.
4. Exécutez la commande suivante :

© Copyright IBM Corp. 2007, 2014

```
run ant -Dwar.file=lavaJSR168.war
```

Elle crée un sous-répertoire appelé tempDir.

5. A l'aide de WebLogic Workshop, importez tempDir en tant que projet Web de portail.

Déploiement du fichier lavaJSR168.war dans le serveur de portail WebLogic

Une fois le fichier `lavaJSR168.war` converti, vous pouvez le déployer.

Procédure

1. Installez le serveur de portail WebLogic, si ce n'est pas déjà fait.
2. Créez une nouvelle instance de domaine de portail et démarrez le nouveau serveur de domaine du portail.
3. A l'aide de WebLogic Workshop, créez une nouvelle application de portail.
4. Dans la nouvelle application de portail, importez le projet Web de portail tempDir que vous avez créé dans la section précédente.

   Avant de continuer, vérifiez que le fichier `lavaJSR168.war` se trouve dans le répertoire WEB-INF/lib du projet Web et que le fichier `cqjsr168.jar` figure bien dans la section portlet.
5. Lancez la Console d'administration de portail et connectez-vous.

   Avant de continuer, vérifiez que l'application Web sélectionnée dans la console est la bonne. Vérifiez également que le tableau de bord s'affiche sous les portlets dans le noeud des modules.
6. Créez une nouvelle page et ajoutez le tableau de bord.
7. Créez un nouveau portail.
8. Sous le nouveau portail, créez un nouveau bureau.
9. Sous le nouveau bureau, créez un livre blanc et ajoutez-y la page que vous avez créée à l'étape précédente.
10. Affichez le bureau pour lancer l'application de portail.
11. Dans le portlet de tableaux de bord, passez en mode édition et connectez-vous aux serveurs IBM Cognos Real-time Monitoring en cours d'exécution.

    Cette action affiche les mesures disponibles.
12. Sélectionnez les mesures que vous souhaitez afficher dans la page de portail.
13. Cliquez sur OK.

    Cette action affiche les objets de tableau de bord qui sont disponibles pour le portlet.

Intégration dans le serveur de portail IBM WebSphere

Vous pouvez procéder à l'intégration d'IBM Cognos Real-time Monitoring dans le serveur de portail IBM WebSphere.

Pour effectuer la procédure suivante, lisez la documentation du portail WebSphere.

Procédure

1. Installez le serveur de portail WebSphere, si ce n'est déjà fait.
2. Connectez-vous à la console d'administration de portail.
3. Dans Gestion de portail > Modules Web, déployez le fichier `lavaJSR168.war`.
4. Créez un nouveau serveur de portail virtuel et donnez-lui un nom approprié.

5. Dans le serveur de portail virtuel, créez une page.
6. Pour configurer la mise en page :
   - Sélectionnez une mise en page en deux colonnes.
   - Ajoutez le portlet de tableaux de bord.
7. Lorsque vous avez terminé, le portail affiche le nom de la page sous la forme d'un lien dans la colonne de gauche. Lorsque vous cliquez sur le lien Page, la colonne de droite affiche le portlet de tableaux de bord.
8. Cliquez sur le lien Page et passez en mode édition du portlet de tableaux de bord.
9. Accédez aux serveurs IBM Cognos Real-time Monitoring en saisissant le nom de l'utilisateur, le mot de passe et l'URL de l'instance en cours d'exécution.

   Cette action affiche les mesures disponibles.
10. Sélectionnez les mesures que vous souhaitez afficher dans la page de portail.
11. Cliquez sur OK.

    Cette action affiche les objets de tableau de bord qui sont disponibles pour le portlet.

Intégration dans le serveur de portail JBoss

Vous pouvez procéder à l'intégration d' IBM Cognos Real-time Monitoring dans le serveur de portail JBoss.

Procédure

1. Installez le serveur de portail JBoss, si ce n'est déjà fait.
2. Arrêtez le serveur d'applications JBoss, s'il est en cours d'exécution.
3. Copiez le fichier `lavaJSR168.war` dans le répertoire de déploiement de l'arborescence d'installation du serveur d'applications JBoss.
4. Redémarrez le serveur d'applications JBoss :

   ...\$JBOSS_HOME$\bin\run.bat (Windows)

   .../$JBOSS_HOME$/bin/run.sh (UNIX)
5. Affichez la page Vue dans un navigateur.

   Par exemple, l'URL de cette page peut être http://[hôte]:[port]/portal/portal/default/[nom de page].

   Vous pouvez personnaliser le nom de la page en modifiant l'élément *<nom de page>* dans le fichier `celequest-objects.xml` du pack WAR. Le nom par défaut est `celequestpage`.
6. Passez en mode édition du portail de tableau de bord.
7. Accédez aux serveurs IBM Cognos Real-time Monitoring en saisissant le nom d'utilisateur, le mot de passe et l'URL de l'instance en cours d'exécution.

   Cette action affiche les mesures disponibles.
8. Sélectionnez les mesures que vous souhaitez afficher dans la page de portail.
9. Cliquez sur OK.

   Cette action affiche les objets de tableau de bord qui sont disponibles pour le portlet.

Instances de portlet multiples sur un serveur JBoss Portal

Vous pouvez créer plusieurs instances de portlet d'un tableau de bord sur le serveur de portail JBoss. Vous pouvez créer plusieurs pages de portail et afficher plusieurs ensembles de tableaux de bord sur ces pages.

Création de plusieurs instances de CelequestDashboardPortletInstance

Vous pouvez créer des instances de portlet d'un tableau de bord sur le serveur de portail JBoss.

Procédure

1. Connectez-vous à JBoss Portal avec le nom d'utilisateur et le mot de passe de l'administrateur.
2. Choisissez la page Portail d'administration sur l'onglet.
3. Cliquez sur le lien Gérer les instances.
4. Sélectionnez LavaDashboardPortletInstance.
5. Donnez un autre nom à l'instance et cliquez sur l'option permettant de créer une nouvelle instance.
6. Répétez les étapes 4 et 5 pour créer d'autres instances de LavaDashboardPortletInstance.

Création de plusieurs pages sur le serveur de portail JBoss :

Vous pouvez créer plusieurs pages de portail sur le serveur de portail JBoss.

Procédure

1. Cliquez sur le lien de gestion du portail.
2. Cliquez sur le dossier racine de l'arborescence.
3. Cliquez sur la page par défaut dans l'arborescence racine.
4. Créez une nouvelle page (par exemple, celPage1) dans la sous-fenêtre gauche, puis cliquez sur cette page.
5. Maintenant, associez la nouvelle page à l'instance LavaDashboardPortletInstance située au centre de la page.
6. Cliquez sur le lien Aperçu de la page créée à l'étape 5 et modifiez le portlet de tableaux de bord IBM Cognos Real-time Monitoring. Par exemple, ajoutez les objets de tableau de bord dans la page.
7. Répétez les étapes 4, 5 et 6 pour créer des portlets de tableaux de bord Cognos Real-time Monitoring pour chaque instance de LavaDashboardPortletInstance.

Chapitre 25. Processus

Un processus est une suite d'étapes qui accomplissent une tâche. Une transaction réelle effectuée via un processus est une instance de processus.

L'exemple ci-dessous décrit un processus de quatre étapes qui crée une demande d'approbation. Une demande est créée, vérifiée, puis approuvée ou rejetée. Une transaction réelle, telle qu'une demande d'approbation spécifique, est une instance de processus.

Figure 27. Processus en quatre étapes

Les systèmes BPM (Business Process Management, gestion de processus métier) créent et gèrent les processus métier et les instances. Lorsqu'un système BPM gère une instance de processus, il envoie des détails sur chaque étape de la transaction à IBM Cognos Real-time Monitoring, qui génère ensuite des statistiques sur le processus global. Par exemple, le système peut déterminer la durée moyenne requise pour mener le processus à terme, le gain de temps réalisé par le processus avec le temps, le pourcentage de demandes rejetées et la durée de l'étape de vérification.

Fonctionnement des processus

Les détails de l'instance de processus sont transmis à IBM Cognos Real-time Monitoring sous la forme d'événements. Le serveur Cognos Real-time Monitoring compile ces détails dans des vues ou des cubes pour générer des statistiques. Cognos Real-time Monitoring Dashboard présente le processus dans un diagramme et les statistiques dans un tableau. D'autres objets peuvent afficher d'autres mesures, comme les graphiques qui présentent des statistiques sur la durée.

Figure 28. Système de gestion de processus envoyant des détails sur une instance à IBM Cognos Real-time Monitoring

Cognos Real-time Monitoring Dashboard permet également d'envoyer une requête au système BPM pour obtenir des détails sur une instance de processus donnée, comme son emplacement au sein du processus. Lorsqu'un utilisateur émet une telle requête, Cognos Real-time Monitoring Dashboard interroge une table de correspondance sur les serveurs Cognos Real-time Monitoring. Cette table

© Copyright IBM Corp. 2007, 2014

interroge le système BPM pour obtenir des détails sur l'instance concernée. Les résultats sont ensuite renvoyés à Cognos Real-time Monitoring Dashboard où ils sont affichés.

Figure 29. Tableau de bord interrogeant une table de correspondance et IBM Cognos Real-time Monitoring interrogeant le système de gestion de processus.

Le système de gestion de processus génère un fichier de définition de processus qui décrit le processus au format XML. Dans IBM Cognos Real-time Monitoring Workbench, une définition du processus associe ce fichier de définition à une vue ou un cube agrégé(e) et, éventuellement, à une table de correspondance de recherche. Cognos Real-time Monitoring Dashboard utilise cet objet comme source pour créer un diagramme de processus et pour identifier la vue ou le cube agrégé correspondant ainsi que la table de correspondance de recherche.

Figure 30. Système de gestion de processus envoyant le fichier de définition de processus à IBM Cognos Real-time Monitoring.

Création et utilisation des processus

Pour créer et utiliser des processus, le système de processus métier externe doit effectuer plusieurs tâches.

Ces tâches sont les suivantes :

- Générer un fichier de définition de processus dans un format XML reconnu par IBM Cognos Real-time Monitoring.
- Publier des statistiques sur les étapes du processus, sous la forme de données d'événements dans les serveurs Cognos Real-time Monitoring.
- Fournir une interface pour les requêtes d'instance de processus émises par les tables de correspondance Cognos Real-time Monitoring. Cette tâche est facultative.

Procédure

1. Créez un agent (le cas échéant) et une table de flux de données destinée à recevoir les événements de processus envoyés par le système BPM.
2. Créez une vue ou un cube par processus.
3. (Facultatif) Créez une table de correspondance pour interroger le système BPM.
4. Créez un objet de définition de processus.
5. Créez un diagramme de processus.

Tables de flux de données

Les tables de flux de données reçoivent et agrègent les statistiques de l'instance de processus. Les données d'événement sont reçues dans une table de flux de données, généralement via un agent. Les vues agrégées sont basées sur la table de flux de données.

Pour plus d'informations sur ces composants, reportez-vous aux rubriques suivantes :

- Chapitre 4, «Agents», à la page 25
- Chapitre 10, «Flux de données», à la page 85
- Chapitre 36, «Vues», à la page 363
- Chapitre 8, «Cubes», à la page 63

Les conditions requises suivantes s'appliquent également :

- Un système BPM ne peut contenir qu'un agent et une table de flux de données.
- Il doit exister une vue ou un cube par processus. Utilisez la clause WHERE pour distinguer les événements de processus des autres processus dans la table d'événements, comme "Nom Processus"='Approbation Requête'. Pour plus d'informations, reportez-vous aux définitions des processus dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Recherche de table de correspondance

La table de correspondance génère une requête dans le BPM chaque fois qu'un utilisateur IBM Cognos Real-time Monitoring Dashboard crée une requête ad hoc sur une instance de processus.

Pour plus d'informations, voir Chapitre 20, «Tables de correspondance», à la page 265.

Lorsque la table de correspondance reçoit une requête envoyée par Cognos Real-time Monitoring Dashboard, elle commence par rechercher les détails dans le cache de la table de correspondance. Si l'instance ne figure pas dans le cache, la table interroge le BPM. N'oubliez pas de définir un calendrier d'invalidation raisonnable pour votre entreprise ou désactivez le cache si les requêtes doivent renvoyer les informations les plus à jour sur l'instance de processus.

Fichier de définition de processus

Pour créer une définition de processus, vous avez besoin d'un fichier de définition de processus qui décrit votre système BPM. Ce fichier doit être conforme à XPDL (XML Process Definition Language) 2.0.

IBM Cognos Real-time Monitoring prend en charge les éléments suivants :

- Plusieurs éléments workflowProcess dans un document XPDL.
- Des icônes par défaut pour tous les types de passerelle et d'événement.

Tous les documents XPDL doivent comporter la mention 2.0 dans l'élément Package.PackageHeader.XPDLVersion.

Il existe des exceptions à la norme XPDL 2.0 lors du rendu des diagrammes de processus par Cognos Real-time Monitoring.

- Les éléments ActivitySet, ainsi que les activités et les transitions définies au sein d'un élément ActivitySet, apparaissent avant tous les autres objets d'un ensemble de tâches. (Les ensembles d'activité apparaissent en premier, car ils sont censés dépeindre un sous-processus.) Les autres objets renvoient des diagrammes en fonction de leur emplacement dans le fichier XPDL, dans lequel plus un objet est spécifié tard dans le fichier de définition XPDL et plus il est susceptible de figurer en première place. L'ordre décroissant des éléments est actuellement le suivant :
  - Transitions
  - Activités
  - Artefacts
  - Associations
  - Messages
  - Couloirs
  - Réserves
- Conformément à la norme XPDL 2.0, l'élément NodeInfoGraphicsInfo est facultatif pour chaque noeud. Toutefois, Cognos Real-time Monitoring requiert un élément NodeInfoGraphicsInfo pour chaque noeud du diagramme de processus. Si plusieurs éléments NodeInfoGraphicsInfo sont associés à un seul noeud, seul l'élément doté de la propriété ToolID "Cognos Real-time Monitoring" est lu. De plus, un élément Coordonnées est attendu avec chaque élément NodeGraphicsInfo. Les coordonnées X et Y doivent correspondre à l'angle supérieur gauche de la boîte englobante du noeud. Spécifiez également la hauteur et la largeur.
- Conformément à la norme XPDL 2.0, l'élément ConnectorGraphicsInfo n'est pas requis pour chaque élément de type de connecteur. Toutefois, si vous ne spécifiez pas d'élément ConnectorGraphicsInfo, une route directe est utilisée entre les noeuds. Les transitions qui définissent un type CONDITION ne renvoient pas de losange conditionnel au point de départ de la transition. Un type de condition OTHERWISE renvoie une barre oblique inversée (\) au début de la transition. Lorsque vous indiquez un élément ConnectorGraphicsInfo, vous devez indiquer les coordonnées du début à la fin et non inversement.
- Les événements s'affichent uniquement tel que spécifié par la norme XPDL 2.0. Les attributs obsolètes servant à indiquer un processus de début ou de fin ne sont pas pris en charge par Cognos Real-time Monitoring.
- Les connecteurs et les noeuds qui contiennent un attribut de nom affichent le texte de cet attribut sous forme de libellé.
- Lorsque vous indiquez des URL en vue de personnaliser des icônes, l'adresse URL peut être spécifiée dans l'attribut Forme d'un élément NodeGraphicsInfo ou dans le sous-élément Icône d'un élément Activité. Vous pouvez soit utiliser une adresse URL complète, soit une adresse URL relative dans Cognos Real-time Monitoring Dashboard. Les icônes personnalisées ne sont pas désaxées pour correspondre à la largeur et à la hauteur définies dans l'élément NodeGraphicsInfo. Les icônes personnalisées conservent leur rapport d'aspect.

Création de définitions de processus

Vous pouvez créer des définitions de processus.

Pour plus de détails sur cette tâche, reportez-vous aux définitions des processus dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Avant de commencer

Avant de créer une définition du processus, vous devez disposer des éléments suivants :

- Fichier de définition de processus. Ce fichier doit être conforme à XPDL (XML Process Definition Language) 2.0.
- Accès en lecture seule de la vue ou du cube qui rassemble les événements de processus.
- Vue ou cube totalisant les données de l'instance de processus. Une colonne répertorie les étapes du processus.

L'accès en lecture seule sur la table de recherche de la table de correspondance est facultatif.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Définition du processus**.
4. Nommez la définition de processus, spécifiez un dossier et fournissez une description. Le dossier par défaut est **Dossiers publics**. Une description est facultative.
5. Cliquez sur **Télécharger le fichier de définition du processus** pour télécharger le fichier de définition du processus généré par BPM.

   Le nom du processus provenant du fichier de définition apparaît dans la zone **Nom du processus** une fois que vous avez chargé le fichier. Pour en savoir davantage, voir «Fichier de définition de processus», à la page 309.
6. Cliquez sur **Sélectionner une source de données** pour sélectionner la vue ou le cube agrégé(e).
7. Utilisez la liste **Colonne Etape** pour sélectionner la colonne Etape. Par exemple, chaque étape du processus peut être identifiée par un nom ou un numéro d'identificateur.
8. Pour créer une recherche de détail, cliquez sur **Sélectionner une source de données**, puis sélectionnez la colonne faisant l'objet de la recherche depuis la liste **Colonne Rechercher**. Vous pouvez saisir un nom dans la zone **Nom descriptif**. Ce nom apparaît dans IBM Cognos Real-time Monitoring Dashboard.
9. Cliquez sur **Enregistrer**.

   Vous pouvez désormais utiliser cette définition pour créer des diagrammes de processus.

Création de diagrammes de processus

Vous pouvez créer des diagrammes de processus dans IBM Cognos Real-time Monitoring Dashboard à l'aide des définitions de processus que vous avez créées dans IBM Cognos Real-time Monitoring Workbench.

Pour plus de détails sur cette tâche, reportez-vous à la section des diagrammes de processus du *Guide d'utilisation d'IBM Cognos Real-time Monitoring Dashboard*.

Avant de commencer

Pour créer un diagramme de processus, les définitions de processus doivent déjà être définies dans IBM Cognos Real-time Monitoring Workbench. De plus, vous devez disposer d'un accès en lecture seule sur la vue ou le cube agrégé(e) qui fournit les statistiques.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, ouvrez le gestionnaire **Diagrammes de processus**.
2. Cliquez sur **Créer un diagramme**.
3. Sélectionnez la définition du processus telle que définie dans IBM Cognos Real-time Monitoring Workbench.
4. Sélectionnez les colonnes source à afficher sous la forme de statistiques.
5. Attribuez un nom au diagramme de processus et enregistrez-le.

   IBM Cognos Real-time Monitoring Dashboard affiche instantanément le diagramme de processus. Les statistiques sont mises à jour à mesure que les événements concernant le processus sont enregistrés.

Chapitre 26. Tableaux récapitulatifs

Les tableaux récapitulatifs décrivent le contenu d'une vue et présentent ces informations dans un rapport joint à un message d'alerte ou envoyé par un système externe.

Par exemple, lorsque le nombre d'unités disponibles pour un produit est bas et que le réapprovisionnement est en retard, une alerte peut avertir le responsable des achats et un tableau récapitulatif joint à l'alerte peut répertorier d'autres fournisseurs de ce produit. Les tableaux récapitulatifs sont joints pour tous les abonnés à l'alerte associée. Pour en savoir davantage, voir Chapitre 5, «Alertes», à la page 43.

Il existe deux types de tableaux récapitulatifs :

- Les tableaux récapitulatifs internes qui sont la représentation visuelle des informations de la vue lorsque l'alerte a généré le tableau récapitulatif. La présentation est une table en texte brut ou au format HTML, qui contient l'ensemble des informations incluses dans la vue.
- Les tableaux récapitulatifs externes (tiers) sont produits par des systèmes externes d'élaboration de rapports. Ils affichent un rapport en fonction des données qui leur sont transmises lorsque l'utilisateur clique sur un lien pointant vers le système externe. Ce système a en charge la génération et la présentation du rapport.

Création de tableaux récapitulatifs

Vous pouvez créer des tableaux récapitulatifs afin de fournir des informations sur un événement.

Pour plus d'informations sur les attributs des tableaux récapitulatifs, voir «Attributs des tableaux récapitulatifs», à la page 314 ou «Attributs des tableaux récapitulatifs externes», à la page 314.

Avant de commencer

Vous devez disposer d'autorisations spécifiques pour créer un tableau récapitulatif :

- Autorisation de création. Pour en savoir davantage, reportez-vous aux sections Chapitre 23, «Autorisations», à la page 295 et «Autorisation de création d'objets», à la page 299.
- Autorisation de lecture et écriture sur l'activité métier qui contient le tableau récapitulatif. Pour en savoir davantage, voir Chapitre 23, «Autorisations», à la page 295.
- Autorisation de lecture seule sur la vue qui alimente le tableau récapitulatif.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Modélisateur de scénarios**.
2. Sous **Activités métier**, cliquez sur le scénario existant qui va contenir le tableau récapitulatif.
3. Cliquez sur l'onglet **Tableaux récapitulatifs**.

© Copyright IBM Corp. 2007, 2014

4. Cliquez sur **Nouveau tableau récapitulatif**.
5. Sélectionnez le type de tableau récapitulatif à créer.

   Les tableaux récapitulatifs se présentent sous la forme de tables HTML.

   Les tableaux récapitulatifs externes (tiers) sont définis et produits par des systèmes externes d'élaboration de rapports, en fonction des données qui leur sont transmises.

   L'option Tableaux récapitulatifs externes n'est disponible que lorsque des liens externes ont été définis. Pour plus d'informations, voir le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
6. Complétez les zones d'attributs dans la fenêtre Créer un tableau récapitulatif.
7. Enregistrez le tableau récapitulatif comme activé.

   Vous pouvez également créer un tableau récapitulatif lors de la création ou de la modification d'une alerte. Si tel est le cas, le tableau récapitulatif est lié à cette alerte. Pour en savoir davantage, voir Chapitre 5, «Alertes», à la page 43.

Attributs des tableaux récapitulatifs

Les tableaux récapitulatifs se présentent sous la forme de tables HTML dotées d'attributs particuliers.

Tableau 86. Attributs des tableaux récapitulatifs

Attribut	Description
Nom	Identifie le tableau récapitulatif. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être unique parmi les tableaux récapitulatifs d'un même scénario. Pour plus de détails, voir «Espace-noms d'objets», à la page 281.
Description	Description facultative qui peut contenir tous les caractères de texte.
Etat	Détermine si le tableau récapitulatif est activé ou désactivé. Lorsque le scénario correspondant est désactivé, il est impossible d'activer le tableau récapitulatif. Le scénario doit impérativement être activé pour que le tableau récapitulatif le soit également.
Vue	Vue métier dont le rapport tire ses données. Pour en savoir davantage, voir Chapitre 36, «Vues», à la page 363.

Attributs des tableaux récapitulatifs externes

Les tableaux récapitulatifs externes affichent les rapports en fonction des données qui leur sont transmises lorsque l'utilisateur clique sur un lien pointant vers le système externe (tiers).

Ce système a en charge la génération et la présentation du rapport. La définition du tableau récapitulatif externe identifie le lien externe et les informations de vue à transmettre au système externe que ce dernier va ensuite utiliser pour identifier le rapport à présenter. Par exemple, un rapport externe peut présenter un fichier PDF qui contient la description complète d'un produit identifié dans une alerte.

Tableau 87. Attributs des tableaux récapitulatifs externes

Attribut	Description
Nom du tableau récapitulatif	Identifie le tableau récapitulatif. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être unique parmi les tableaux récapitulatifs d'un même scénario. Pour plus de détails, voir «Espace-noms d'objets», à la page 281.

Tableau 87. Attributs des tableaux récapitulatifs externes (suite)

Attribut	Description
Etat	Détermine si le tableau récapitulatif est activé ou désactivé. Lorsque le scénario correspondant est désactivé, il est impossible d'activer le tableau récapitulatif. Le scénario doit impérativement être activé pour que le tableau récapitulatif le soit également.
Description	Description facultative qui peut contenir tous les caractères de texte.
Données de la vue	Vue métier dont le rapport tire ses données. Cette vue contient les données de colonne à envoyer au rapport externe. Pour en savoir davantage, voir Chapitre 36, «Vues», à la page 363.
Nom du rapport	Nom du rapport dans le système externe. Il s'agit de l'élément DocName mentionné dans l'URL qui communique avec le système de rapport externe : `http://localhost.com?`**`DocName=<Report Name>`**
Paramètres de rapport	Paramètres à transmettre au système externe. Chaque paramètre correspond à une colonne dans la vue. Le tableau récapitulatif remplace la valeur de chaque colonne nommée dans l'URL. Par exemple, l'URL se présente comme suit : `...?DocName=<Report Name>`**`&Parameter1=PROD_ID&...`** Elle est similaire à celle ci-dessous lorsqu'elle est envoyée au système externe : `...?DocName=<Report Name>`**`&product="product_id"&...`**
Arguments de rapport	Cette sous-fenêtre indique les valeurs pouvant être affectées aux arguments dans l'adresse URL spécifiée dans le lien externe. Prenez, par exemple, le cas où le lien externe définit une adresse URL de base spécifique pour un rapport IBM Cognos et que l'adresse URL de base nécessite les arguments suivants : **ui.tool**=CognosViewer&**ui.object**=content/folder[@name='Demo Report']/report[@name='Inventory Levels by Retailers']**ui.action**=run &**run.outputFormat**=PDF L'URL de base est la suivante : http://nom_serveur/ibmcognos/cgi-bin/cognos.cgi?b_action=xts.run&m=portal/launch.xts& Lorsque le rapport associé est appelé à partir de l'objet de tableau de bord avec les valeurs définies pour les arguments dans la sous-fenêtre des arguments de rapport, la demande d'adresse URL ressemble à ce qui suit : http://nom_serveur/ibmcognos/cgi-bin/cognos.cgi?b_action=xts.run &m=portal/launch.xts&**ui.tool**=CognosViewer**& ui.object**=content/ folder[@name='Demo Report']/report[@name='Inventory Levels by Retailers']&**ui.action**=run&**run.outputFormat**=PDF
Afficher le lien	Affiche l'URL complète qui sera mentionnée dans le message d'alerte. Il s'agit du lien permettant d'accéder au système de rapport externe.

Les URL utilisées pour communiquer avec le système de rapport externe commencent par la chaîne définie pour le rapport dans la liste Liens externes de l'onglet **Plan de travail** dans Cognos Real-time Monitoring Workbench. Pour plus d'informations, voir le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Vues de tableau récapitulatif

Les tableaux récapitulatifs extraient leurs informations de la vue métier sur laquelle l'alerte est basée ou de toute vue provenant du même flux de données.

Prenons l'exemple d'une règle qui a généré l'alerte client basée sur la vue InventoryLow. Une autre vue, AvailableSuppliers, ajoute du contexte en indiquant d'autres fournisseurs. Le tableau récapitulatif joint à l'alerte peut tirer des informations de l'une ou l'autre de ces vues. De plus, comme la vue ShippingNotices, est générée à partir de la même source d'événements, vous pouvez également en extraire des informations. Toutefois, vous ne pourrez pas extraire des informations de la vue OrderDetails, car elle appartient à un autre flux de données.

Figure 31. Vues de tableau récapitulatif

Pour en savoir plus sur les informations fournies dans la vue du tableau récapitulatif, voir «Filtrage des tableaux récapitulatifs», à la page 51.

Notez que la vue du tableau récapitulatif ne peut pas être une jointure synchronisée. Pour en savoir davantage, voir «Jointures synchronisées», à la page 366.

Chapitre 27. Rôles

Les rôles définissent les ensembles minimaux de droits associés aux utilisateurs.

Ils permettent d'attribuer rapidement les mêmes autorisations à un objet ou une classe d'objets. Par exemple, un rôle d'opérateur peut attribuer toutes les autorisations à des agents mais pas à des flux de données ni à des objets d'activité métier, tandis qu'un rôle de développeur d'application peut posséder toutes les autorisations sur l'ensemble des objets, à l'exception des agents.

Les utilisateurs peuvent avoir aucun, un ou plusieurs rôles. Pour connaître les rôles d'un utilisateur, modifiez le compte utilisateur et affichez l'onglet Détails de l'utilisateur. Pour voir les rôles qui sont les vôtres, cliquez sur **Paramètres du compte** et affichez l'onglet. Pour en savoir davantage, voir «Onglet Détails de l'utilisateur», à la page 355.

L'autorisation accordée à un utilisateur d'effectuer une opération donnée correspond au niveau maximum de toutes les autorisations associées aux rôles de cet utilisateur et à chacune des autorisations attribuées à l'utilisateur sur l'objet. Prenons le cas d'un utilisateur avec deux rôles : le premier avec un accès en lecture seule à la classe de vues, le second avec un accès en lecture et en écriture. L'autorisation maximum sur les vues de cet utilisateur est Lecture et écriture. Par conséquent, il peut modifier des vues.

De même, si un utilisateur dispose du rôle Lecture seule sur l'ensemble des vues mais qu'il a personnellement reçu le rôle Lecture et écriture sur une vue particulière, il peut modifier cette vue.

Comme tous les autres objets IBM Cognos Real-time Monitoring, les rôles sont protégés par des autorisations. Seuls les utilisateurs qui possèdent certaines autorisations sur les rôles sont habilités à les modifier ou à les supprimer. Par exemple, pour ajouter des utilisateurs à un rôle, vous devez disposer de l'autorisation d'accès en lecture et en écriture sur ce rôle.

La liste **Rôles** de la **Console d'administration** affiche les rôles en cours sur le système.

Pour plus d'informations sur les niveaux d'accès spécifiques des rôles, voir Chapitre 23, «Autorisations», à la page 295.

Création de rôles

Les rôles définissent le nombre minimum d'autorisations associées aux utilisateurs.

Avant de commencer

Pour créer un rôle, vous devez disposer de l'autorisation de création sur les rôles. Pour chaque classe d'objets, vous pouvez attribuer jusqu'à l'autorisation la plus élevée que vous possédez sur cette classe.

Par exemple, si vous possédez l'autorisation de création sur une classe, vous pouvez lui attribuer les autorisations suivantes :

© Copyright IBM Corp. 2007, 2014

- Aucun accès
- Lecture seule
- Lecture et écriture
- Création

Procédure

1. Dans l'onglet **Console d'administration**, cliquez sur **Rôles** pour afficher la liste des rôles actuellement définis.
2. Cliquez sur **Nouveau rôle**.
3. Précisez les attributs du rôle, attribuez les autorisations d'accès et désignez les membres du rôle.

Attributs de rôle

Chaque objet de rôle est associé à des attributs.

Tableau 88. Attributs de rôle

Attribut	Description
Nom	Indique le nom du rôle. Ce nom doit être unique parmi tous les objets du dossier racine, notamment les activités métier, les utilisateurs, les rôles, les objets de tableau de bord, etc.
Description	Description facultative qui peut contenir tous les caractères de texte.
Droits d'accès	Autorisations pour chaque classe d'objets attribuée à ce rôle. Il s'agit des autorisations minimum. Un utilisateur peut posséder une autorisation supérieure attribuée individuellement à un objet ou une classe d'objets.
Membres	Utilisateurs associés à ce rôle.

Onglet Droits d'accès

Les droits d'accès déterminent le niveau d'accès d'un utilisateur ou d'un rôle sur un objet. Ils peuvent être attribués à une classe d'objets ou à un objet spécifique.

Dans cet onglet, vous pouvez attribuer des autorisations aux éléments suivants :

- Activités métier
- Vues, cubes et dimensions
- Tables de correspondance et flux de données
- Utilisateurs
- Rôles
- Agents
- Propriétés générales du système
- Tableaux de bord
- Objets de tableau de bord
- Processus externes

Les nouveaux utilisateurs et nouveaux rôles ne disposent d'aucun accès sur les objets. Par conséquent, les utilisateurs et les rôles ne peuvent recevoir et voir que les alertes et les tableaux récapitulatifs associés à un abonnement obligatoire.

Pour plus d'informations sur l'attribution des autorisations, voir Chapitre 23, «Autorisations», à la page 295.

Chapitre 28. Règles

Les règles contrôlent les activités métier en analysant les vues métiers qui recherchent des mesures répondant à certains critères.

Les conditions de règle sont des formules du type de celles qu'on rencontre dans un tableur. Elles évaluent des mesures métier changeantes, qui recherchent des conditions exceptionnelles. Lorsqu'une condition est détectée, une alerte de cet incident est envoyée au personnel concerné.

Figure 32. Règle contrôlant une vue

Vous pouvez créer des règles qui envoient des alertes chaque fois que la condition existe (FIRE), qui envoient des alertes une seule fois et ignorent les événements suivants tant que la condition initiale n'est pas résolue (RAISE) ou qui réinitialisent (LOWER) les règles précédemment émises.

Dans les versions précédentes, la règle ne pouvait contrôler que les vues obtenues à partir de la vue par défaut du scénario. Cette restriction n'existe plus.

Création de règles

Vous pouvez créer des règles.

Avant de commencer

Pour ce faire, vous devez disposer de l'autorisation de création sur les activités métier, de l'autorisation de lecture et écriture sur l'activité métier qui contient la règle ainsi que de l'autorisation de lecture seule sur la vue qui alimente la règle.

Pour plus d'informations sur les autorisations, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Modélisateur de scénarios**.
2. Sous **Activités métier**, cliquez sur le scénario existant qui va contenir la règle.
3. Ouvrez un scénario.
4. Dans l'onglet **Règles**, cliquez sur **Nouvelle règle**.

© Copyright IBM Corp. 2007, 2014

5. Choisissez la source de données que la règle doit contrôler.
6. Spécifiez les attributs de la règle de la fenêtre Créer une règle. De plus :
   - Si le scénario a une vue par défaut, c'est celle-ci qui s'affiche par défaut. Pour choisir une autre source, cliquez sur **Sélectionner une source de données**.
   - S'il s'agit d'une vue, choisissez la vue.
   - S'il s'agit d'un cube, choisissez le niveau de dimension. Vous pouvez également appliquer un filtre qui restreint encore davantage les données que la règle va contrôler.

   Si la source contient des données, celles-ci fournissent un échantillon de ce à quoi vous pouvez vous attendre. Lorsque la source est vide, la fenêtre affiche uniquement les noms des colonnes et le message Aucune donnée disponible.

   Pour identifier l'alerte à activer lorsque la condition de règle est satisfaite, sélectionnez une alerte existante en cliquant dessus dans la zone Règle en vigueur ou cliquez sur **Suivant** pour accéder à la fenêtre Définition de l'alerte.
7. Enregistrez la règle avec l'état activé.

Clonage de règles

Vous pouvez cloner et modifier des règles existantes.

Avant de commencer

Pour ce faire, vous devez disposer de l'autorisation de création sur les activités métier, de l'autorisation de lecture et écriture sur l'activité métier qui contient la règle ainsi que de l'autorisation de lecture seule sur la vue qui alimente la règle.

Pour plus d'informations sur les autorisations, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Modifiez la règle que vous souhaitez cloner.
2. Modifiez le nom de la règle et les attributs qui doivent différer de ceux de la règle d'origine.
3. Cliquez sur **Enregistrer sous Nouvelle règle**.

Attributs de règle

Chaque règle est associée à des attributs.

Tableau 89. Attributs de règle

Attribut	Description
Source de données actuelle	Identifie la vue ou le cube que la règle surveille. Cliquez sur **Sélectionner une source de données** pour modifier la source de données de la règle.
Nom de la règle	Identifie l'objet règle. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être unique parmi les règles d'un même scénario. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Spécifie si la règle est activée (réception de nouvelles données) ou désactivée. Lorsque le scénario correspondant est désactivé, il est impossible d'activer la règle. Le scénario doit impérativement être activé pour que la règle le soit également.

Tableau 89. Attributs de règle (suite)

Attribut	Description
Description	Description facultative qui peut contenir tous les caractères de texte.
Condition	Formule vraie ou fausse, du type de celle qu'on trouve dans un tableur, qui est exécutée sur la vue métier associée. Lorsque la formule est vérifiée (True), la condition est remplie et le système envoie l'alerte correspondante. Pour en savoir davantage, voir «Conditions de règle». Lorsque la condition existe, une action est prise. Les règles peuvent envoyer des alertes chaque fois que la condition est vérifiée (lancer), envoyer des alertes une seule fois et ignorer les événements ultérieurs jusqu'à ce que la condition initiale soit rétablie (augmenter) ou réinitialiser (diminuer) les règles précédemment émises. Pour en savoir plus sur les actions prises, voir «Actions de règle», à la page 324.
Correspondant à	N'envoie l'alerte que lorsque la condition est vérifiée pendant la durée spécifiée. Aucune alerte n'est émise si la condition n'est plus vérifiée avant l'expiration du délai. Si vous omettez la durée, le système envoie l'alerte dès que la condition est détectée. Pour spécifier une valeur, procédez au choix comme suit : • Saisissez un nombre indiquant la longueur du délai. • Indiquez le nom d'une colonne dans la vue qui contient le nombre. Lorsque vous utilisez cette option, la règle récupère la valeur à partir de l'événement dans la vue. Dans des fuseaux horaires appliquant une heure d'été, la durée des jours, mois et années est ajustée en conséquence. Même si un jour compte normalement 24 heures, il peut durer 23 ou 25 heures selon le moment.
Simple	L'onglet **Simple** permet d'indiquer l'alerte à envoyer lorsque la condition de règle est satisfaite.
Avancé	L'onglet **Avancé** vous permet de sélectionner plusieurs options d'alerte pour la règle. Outre la sélection de l'alerte à utiliser, vous pouvez spécifier le type d'action à prendre lorsque la condition est remplie, gérer les états d'alerte liées à une colonne spécifique de la vue, ou consolider plusieurs messages pour un même événement.

Conditions de règle

Une condition de règle est une formule qui teste la ligne de la vue métier associée et qui recherche une condition spécifique.

Lorsque la condition est remplie, l'action de la règle active une alerte. Les formules peuvent rechercher tout simplement une valeur dans une colonne de la vue, comme Status='Resolved', ou être des expressions booléennes complexes qui comprennent des fonctions, des opérateurs et des parenthèses, comme (Status='Resolved' OR Status='Assigned') AND UPPER(cust_tier)='HIGH'.

Une condition de règle peut contenir un nombre illimité de références de colonne, d'opérateurs et de fonctions. Toutefois, elle doit respecter les exigences suivantes :

- Le résultat de la formule doit être de type booléen et renvoyer une valeur vraie (True) ou fausse (False).
- Toutes les références de colonne doivent figurer dans la vue métier associée.
- Seules les fonctions scalaires (qui ne s'appliquent qu'à une seule ligne d'une vue) peuvent être utilisées. Pour afficher la liste des fonctions disponibles, cliquez sur **Plus de fonctions** lorsque vous saisissez la condition de règle. Pour en savoir davantage, voir Chapitre 15, «Fonctions C-SQL», à la page 133.

Actions de règle

Les règles peuvent avoir l'un des trois effets suivants : elles peuvent envoyer des alertes chaque fois qu'une condition est vérifiée, envoyer des alertes une seule fois et ignorer les événements ultérieurs jusqu'à ce que la condition initiale soit rétablie ou réinitialiser les règles précédemment émises.

- Envoyer des alertes à chaque fois que la condition est remplie (envoyer). Une action "envoyer" transmet l'alerte dès qu'une règle identifie une condition exceptionnelle. Par exemple, dans un centre de support client qui gère les problèmes des clients sous la forme de tickets d'événement, une alerte peut être émise chaque fois qu'un nouveau ticket est ouvert.
- Envoyer des alertes une seule fois et ignorer les événements suivants jusqu'à ce que la condition initiale soit rétablie (lancer). Une action RAISE envoie un message d'alerte lorsque la condition de la règle s'applique mais qu'elle ignore les événements suivants jusqu'au rétablissement de la condition initiale. Cette action est utile pour éviter les alertes multiples dans les situations où la condition de la règle est vraie pour plusieurs événements liés. Par exemple, si un ticket de problème client ouvert est modifié, il est inutile d'émettre une autre alerte pour la modification, même si le second événement reste dans l'état "ouvert".

  Vous pouvez envoyer une seule fois des alertes, à chaque occurrence spécifique de la colonne nommée. Par exemple, pour envoyer une alerte à chaque ouverture d'un ticket de problème, vous pouvez désigner Ticket comme colonne d'occurrences spécifiques. Une alerte est maintenant émise à chaque ouverture d'un ticket, mais elle est ignorée pour tous les événements suivants de ce ticket tant qu'il conserve l'état "ouvert".
- Réinitialiser (LOWER) les règles précédemment activées pour leur permettre d'envoyer des alertes.

Exemples de règle

Lorsqu'une règle RAISE active une alerte, cette dernière n'est pas réactivée. Les règles RAISE suivantes de l'alerte sont ignorées jusqu'à ce que la règle LOWER la réinitialise.

Avec la condition de règle "Status=Open", le premier événement ci-dessous active l'alerte, mais les suivants sont ignorés tant que l'état de l'alerte n'est pas réinitialisé. Par exemple :

```
Ticket Status
------ ------
0703   Open      << Raise
0706   Open      << Ignore
0704   Open      << Ignore
0705   Open      << Ignore
```

Dans cet exemple, vous souhaitez probablement émettre une alerte pour chaque événement d'ouverture. Vous pouvez utiliser une condition spécifique qui active une instance de l'alerte pour chaque occurrence des valeurs dans les colonnes spécifiées :

```
Raise someAlert when Status = 'Open' for unique occurrences
of Ticket.
```

En désignant la colonne Ticket comme colonne spécifique, une instance d'alerte est activée et un message est envoyé pour chaque événement d'ouverture qui n'a aucune instance pour le numéro de ticket spécifié. De même, vous pouvez réinitialiser chaque instance d'alerte individuellement avec des règles Lower spécifiques.

```
Lower someAlert when Status = 'Open' for unique occurrences
of Ticket.
```

Si vous ignorez les zones spécifiques dans la règle Lower, la règle va réinitialiser toutes les instances de l'alerte qui ont été émises.

Règles contrôlant les alertes

En général, les règles analysent les vues métier et recherchent des mesures qui correspondent à certains critères.

Toutefois, les règles peuvent également contrôler les alertes générées qui recherchent des conditions requérant une attente particulière, à l'aide de la fonction IS_RAISED ().

Pour plus d'informations sur la fonction IS_RAISED, voir «IS_RAISED», à la page 165.

Figure 33. Une première condition de règle est remplie

Par exemple, si une alerte envoyée 4 heures plus tôt est toujours active, une autre règle peut le remarquer et envoyer une autre alerte à un échelon supérieur. Dans ces descriptions de règle, l'alerte EscalateAlert est émise lorsque OpenAlert est active pendant au moins 4 heures :

```
Raise OpenAlert when Status
= 'Open'

Reset OpenAlert when Status
<> 'Open'

Raise EscalateAlert when IS_RAISED('OpenAlert')
holds for 4 HOUR.

Reset EscalateAlert when NOT
IS_RAISED('OpenAlert')
```

Alertes spécifiques

Lorsqu'une alerte est générée pour des occurrences uniques de zones, le système garde la trace de chaque alerte grâce à ces valeurs de zone.

Par exemple, la règle suivante émet des alertes et contrôle celles qui sont ouvertes grâce aux valeurs uniques dans la zone Ticket :

```
Raise OpenAlert when Status = 'Open' for
unique occurrences of Ticket
```

Pour gérer correctement cette alerte, la règle contenant la fonction IS_RAISED() requiert la même condition :

```
Raise EscalateAlert when IS_RAISED('OpenAlert') holds
for 4 HOUR
for unique occurrences of Ticket
```

Si vous omettez la condition de zone spécifique, l'alerte EscalateAlert n'est émise que pour la première alerte OpenAlert. De même, vous devez réinitialiser les alertes avec des conditions de règle propres aux mêmes zones, sinon, vous réinitialisez toutes les alertes émises.

```
Reset OpenAlert when Status
<> 'Open' for unique occurrences of Ticket
```

```
Reset EscalateAlert when NOT
IS_RAISED('OpenAlert')
for unique occurrences of Ticket
```

Journal système

Vous pouvez concevoir une règle qui contrôle les messages de journal système dans des conditions d'erreur à priorité élevée et les rapporte aux administrateurs clés.

Pour ce faire, suivez les étapes permettant de contrôler les journaux fournies dans la documentation IBM Cognos Real-time Monitoring Workbench.

Cognos Real-time Monitoring génère des messages qui facilitent la maintenance du logiciel en produisant des rapports analysables par les utilisateurs, les administrateurs système, les ingénieurs d'exploitation et les équipes de développement de logiciels. Pour en savoir plus sur la journalisation système, voir la documentation Cognos Real-time Monitoring Workbench.

Chapitre 29. Salesforce

Vous pouvez consulter et importer des tables à partir d'un système de gestion de la relation client (CRM) Salesforce.

La procédure se déroule en deux étapes. La première consiste à créer un agent Salesforce qui se connecte aux tables Salesforce. La seconde, à utiliser un assistant d'extraction distinct pour importer les événements et les contextes Salesforce.

Les objets Salesforce importés peuvent ensuite être utilisés comme sources de données pour les objets de tableau de bord comme les vues métier et les cubes.

Création d'un agent Salesforce

Un agent Salesforce importe les tables Salesforce à utiliser dans des tables de flux de données et des tables de correspondance.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > **Agent**.
4. Choisissez **Salesforce** comme type d'agent.
5. Entrez un nom dans la zone **Nom** et sélectionnez Activé dans la zone **Etat**.
6. Entrez un **Nom d'utilisateur** et un **Mot de passe**.
7. Cliquez sur **OK**.

   Vous pouvez maintenant utiliser un assistant pour importer des tables de flux de données et de correspondance dans IBM Cognos Real-time Monitoring Workbench.

Importation de tables de correspondance et de flux de données Salesforce

Vous pouvez utiliser l'assistant d'importation d'un objet SalesForce.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Console d'administration**.
2. Sous **Administration**, cliquez sur **Assistants**.
3. Dans l'onglet **Assistants**, cliquez sur **Assistant d'importation d'un objet SalesForce**.
4. Sélectionnez un agent Salesforce.
5. Sélectionnez les utilisateurs à importer dans **Liste des utilisateurs**.
6. Sélectionnez les tables à importer dans la zone **Liste de tables** pour les flux de données et les tables de correspondance, puis cliquez sur **Ajouter**.

   Lorsque vous ajoutez une table, elle disparaît de la zone **Liste de tables** et apparaît dans la liste **Tables sélectionnées**. Pour la faire revenir dans la première liste, cliquez sur **Supprimer**.

© Copyright IBM Corp. 2007, 2014

7. Cliquez sur **OK**.

   Les tables de flux de données et de correspondance sont créées à partir des tables sélectionnées en ajoutant respectivement _events (flux de données) et _context (table de correspondance) au nom de la table indiqué dans **Tables et vues**. Par exemple, si vous importez la table Salesforce Account dans une table de flux de données, elle porte le nom de flux de données Account_event.

   Si une table de la liste **Tables sélectionnées** est utilisée pour une table de flux de données ou de correspondance, vous ne pouvez pas la supprimer de cette liste. Vous devez tout d'abord supprimer la table de flux de données ou de correspondance.

Fonction de mise à plat Salesforce

Lors de la création d'une table de correspondance à partir d'une source Salesforce, les agents associés disposent d'une fonction Flatten qui aplatit une table Salesforce.

Toutefois, la table ne peut être aplatie que si elle représente une structure arborescente, dans laquelle le niveau supérieur correspond à l'élément racine de l'arborescence. Une fois la table aplatie, chaque ligne de la nouvelle table représente un noeud feuille de l'arborescence d'origine.

Prenons l'exemple de la table suivante :

Tableau 90. Table de noeuds

NodeName	NodeID	NodeParentId
A	1	NULL
B	2	1
C	3	1
D	4	2
E	5	2
F	6	5
G	7	5

Cette table peut être représentée par l'arborescence ci-dessous. Cette arborescence compte quatre niveaux, avec l'élément A situé au niveau 0.

Après aplatissement, les lignes de la nouvelle table représentent les noeuds terminaux C, D, F et G.

Tableau 91. Lignes de la table représentant des noeuds terminaux

Niveau 0	Niveau 1	Niveau 2	Niveau 3
A	B	D	NULL
A	B	E	F
A	B	E	G
A	C	NULL	NULL

Syntaxe

Cette fonction respecte la syntaxe suivante :

```
FLATTEN( tableName, idColumnName, parentIdColumnName,
nameColumn, prefixForLevelColumns )
```

Paramètres

Les paramètres de la fonction sont définis comme suit :

- tableName

  Nom de la table à aplatir.
- idColumnName

  Colonne d'ID de la table (c'est-à-dire la colonne à laquelle `parentIdColumnName` est liée). Par exemple, dans la description ci-dessus, la valeur idColumnName est NodeID.
- parentIdColumnName

  Colonne contenant les ID des noeuds parent de chaque ligne. Par exemple, dans la description ci-dessus, le noeud A a l'ID 1 et est parent du noeud B. Par conséquent, l'ID parent du noeud B dans la colonne NodeParentID est 1. De même, le noeud parent du noeud D est le noeud B, et le NodeParentID du noeud D est 2 car l'ID du noeud B est 2.
- nameColumn

  Colonne contenant les données à afficher dans la table aplatie.
- prefixForLevelColumns

  Préfixe qui est ajouté à la colonne. Dans la description ci-dessus, le préfixe est Level.

Remarques

Pour que la fonction Flatten fonctionne correctement, la mise en cache prefetch doit être activée dans votre contexte. Pour plus d'informations sur l'activation de la mise en cache, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

Exemple

L'exemple ci-dessous montre la table avant son aplatissement.

Id	Name	ParentRoleId	RollupDescription
00E50000000nJMHEAY	CEO		CEO
00E50000000nJMIEAY	SVP, Customer Service & Support	00E50000000nJMHEAY	SVP, Customer Service & Support
00E50000000nJMJEAY	SVP, Sales & Marketing	00E50000000nJMHEAY	SVP, Sales & Marketing
00E50000000nJNEEAY	VP, Marketing	00E50000000nJMJEAY	VP, Marketing
00E50000000nJNFEAY	VP, North American Sales	00E50000000nJMJEAY	VP, North American Sales
00E50000000nJNGEAY	Director, Channel Sales	00E50000000nJNFEAY	Director, Channel Sales
00E50000000nJNHEAY	Director, Direct Sales	00E50000000nJNFEAY	Director, Direct Sales
00E50000000nJNIEAY	CFO	00E50000000nJMHEAY	CFO

Voici cette même table après son aplatissement avec la fonction :

Id	now_1	now_2	now_3
00E50000000nJNIEAY	CEO	CFO	CFO
00E50000000nJNJEAY	CEO	COO	COO
00E50000000nJNKEAY	CEO	SVP, Customer Service & Support	Customer Support, International
00E50000000nJNLEAY	CEO	SVP, Customer Service & Support	Customer Support, North America
00E50000000nJNMEAY	CEO	SVP, Customer Service & Support	Installation & Repair Services
00E50000000nJNNEAY	CEO	SVP, Human Resources	SVP, Human Resources
00E50000000nJNOEAY	CEO	SVP, Sales & Marketing	VP, International Sales
00E50000000nJNPEAY	CEO	SVP, Sales & Marketing	VP, Marketing

```
flatten(USERROLE, ID, ParentRoleID, Name, cq_)
```

Fonction de liste de sélection Salesforce

Outre la fonction de mise à plat (Flatten), les agents Salesforce proposent également la fonction de liste de sélection (Picklist). Celle-ci renvoie une colonne contenant les valeurs possibles d'une colonne proposant plusieurs choix dans Salesforce.

Syntaxe

Cette fonction respecte la syntaxe suivante :

```
PICKLIST( tableName,pickListColumn )
```

Paramètres

Les paramètres de la fonction sont définis comme suit :

- tableName
  Nom de la table à partir de laquelle renvoyer la colonne de données.
- pickListColumn
  Colonne de la table spécifiée par tableName, à renvoyer.

Remarques

Pour que la fonction de liste de sélection (Picklist) fonctionne correctement, la mise en cache prefetch doit être activée dans votre table de correspondance. Pour plus d'informations sur l'activation de la mise en cache, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

La colonne à renvoyer doit être une colonne proposant une liste de sélections.

Exemple

La fonction Picklist suivante renvoie les valeurs possibles dans la colonne LeadSource de la table Opportunity.

```
picklist(opportunity,leadsource)
```

La capture d'écran ci-dessous montre la table Opportunity :

AccountId	Name	LeadSource
0015000000G97QHAAZ	Express Logistics SLA	Partner Referral
0015000000G97OKAAZ	Edge Emergency Generator	
0015000000G97QJAAZ	United Oil Emergency Generators	Partner Referral
0015000000G97QJAAZ	United Oil Installations	
0015000000FvUcVAAV	Premier sales div2	
0015000000G97QKAAZ	United Billing div	
0015000000FvUcVAAV	Premier Billing div2	
0015000000G97QKAAZ	United Manufacturing div	

Figure 34. Table Opportunity

La fonction Picklist extrait la colonne LeadSource.

leadsource
Partner Referral
Partner Referral
Partner Referral
Other
Purchased List
Web
Purchased List
Partner Referral

Figure 35. Colonne extraite par la fonction Picklist

Console d'administration Salesforce

Pour obtenir un compte Salesforce.com, vous devez avoir accès à la console d'administration IBM Cognos Real-time Monitoring pour AppExchange.

Pour accéder à cette console, entrez l'adresse suivante dans votre navigateur :
`http://[hôte_local]:[port]/[emplacement_installation]/jsp/salesforceadmin.jsp`

Dans la page qui s'affiche, indiquez les informations requises.

Après avoir cliqué sur **Envoyer**, vous recevez un courrier électronique confirmant l'envoi de votre demande. Le courrier précise également qu'un représentant vous contactera pour confirmer l'envoi et vous informer des étapes suivantes permettant de finaliser la configuration.

Chapitre 30. Connectivité SAP

Vous pouvez consulter et importer des tables de faits sous la forme d'objets ODS, et des cubes sous la forme de cubes OLAP, à partir d'un système SAP.

Le processus se déroule en trois étapes. La première consiste à créer un agent JDBC standard qui se connecte à la base de données SAP. La deuxième, à créer un agent ERP qui utilise un agent JDBC de métadonnées SAP pour accéder au système SAP en tant que type source. Et la troisième, à utiliser un assistant d'extraction tiers pour importer les objets ODS ou les cubes OLAP.

Les objets ODS et les cubes importés sont ensuite utilisés comme sources de données pour les objets de tableau de bord comme les vues métier et les cubes.

Création d'un agent SAP

Un agent SAP est un agent de planification des ressources de l'entreprise (ERP) qui utilise un agent JDBC pour accéder à la base de données du système SAP.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Sélectionnez **Créer** > > **Agent**.
4. Choisissez ERP comme type d'agent.

   Lorsque vous sélectionnez le type d'agent, les paramètres suivants réapparaissent pour montrer les paramètres appropriés au type concerné.
5. Entrez un nom dans la zone **Nom** et sélectionnez Activé dans la zone **Etat**.
6. Dans **Type de source ERP**, sélectionnez SAP-BW.
7. Dans **Agent de données**, sélectionnez Couche de l'application SAP. Vous pouvez également utiliser un agent JDBC pour se connecter à la base de données SAP. Pour plus d'informations sur la création d'un agent JDBC, voir «Agents JDBC (Java Database Connectivity)», à la page 238.
8. Saisissez les **Paramètres de connectivité** appropriés dans les zones suivantes :
   - Nom d'utilisateur
   - Mot de passe
   - Langue
   - ASHost
   - Client
   - Numéro du système
   - Hôte de la passerelle
   - Serveur de la passerelle
9. Cliquez sur **OK**.

   Vous pouvez maintenant utiliser un assistant pour importer des objets ODS (voir «Importation d'objets ODS (Operational Data Store)», à la page 334) et des cubes OLAP (voir «Importation de cubes OLAP», à la page 334).

© Copyright IBM Corp. 2007, 2014

Importation d'objets ODS (Operational Data Store)

Vous pouvez importer des objets ODS (magasins de données) en tant que flux de données ou table de correspondance.

Pour en savoir davantage, reportez-vous aux sections Chapitre 10, «Flux de données», à la page 85 et Chapitre 20, «Tables de correspondance», à la page 265.

Procédure

1. Ouvrez la console d'administration Workbench.
2. Sous **Administration**, cliquez sur **Assistants**.
3. Dans l'onglet **Assistants**, cliquez sur **Assistant d'importation d'un objet ODS**.
4. Sélectionnez un agent ERP existant.
5. Sélectionnez le **Groupe d'information** contenant l'objet ODS de votre choix.
6. Dans la liste déroulante **Table à importer**, sélectionnez la table de faits de votre choix.
7. Indiquez si vous voulez importer l'objet ODS en tant que flux de données ou table de correspondance.
8. Cliquez sur **OK**.

Importation de cubes OLAP

Vous pouvez importer l'assistant d'importation de cube ERP.

Procédure

1. Ouvrez la console d'administration Workbench.
2. Sous **Administration**, cliquez sur **Assistants**.
3. Dans l'onglet **Assistants**, cliquez sur **Assistant d'importation de cube ERP**.
4. Sélectionnez un agent ERP.
5. Sélectionnez le cube à importer.
6. Dans la zone **Importer en tant que**, acceptez la valeur par défaut (les mêmes noms que dans SAP) ou saisissez un autre nom identifiant le cube importé dans le plan de travail.
7. Cliquez sur **OK**.

Chapitre 31. Scénarios

Un scénario est un ensemble de règles, d'alertes et de tableaux récapitulatifs qui identifient des conditions métier exceptionnelles.

Les règles du scénario correspondent aux tests qui déterminent à quel moment une condition exceptionnelle existe ou n'existe plus.

Supprimer ou désactiver un scénario supprime ou désactive les règles, les alertes et les tableaux récapitulatifs qu'il contient.

Figure 36. Activité métier avec deux scénarios

Chaque scénario est associé à des attributs.

Tableau 92. Attributs de scénario

Attribut	Description
Etat du dossier	Spécifie si la règle est activée (réception de nouvelles données) ou désactivée. Lorsqu'un scénario est désactivé, l'ensemble de ses règles, alertes et tableaux récapitulatifs l'est également. Lorsque l'activité métier qui contient le scénario est désactivée, vous ne pouvez pas activer ce scénario. L'activité métier doit être activée pour pouvoir en activer le scénario.
Nom du scénario	Identifie l'objet du scénario. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être unique parmi les scénarios. Pour plus de détails, voir «Espace-noms d'objets», à la page 281.
Description	Description facultative du scénario qui peut contenir tous les caractères de texte.
Vue par défaut	Identifie la vue par défaut (voir Chapitre 36, «Vues», à la page 363) que les règles de ce scénario contrôlent et qui est rapportée dans les tableaux récapitulatifs. Lorsque vous cliquez sur **Aucune vue par défaut**, vous devez choisir une vue pour les règles et les tableaux récapitulatifs au moment de leur création. Lorsque vous cliquez sur **Vue existante**, vous avez la possibilité de choisir une vue existante dans IBM Cognos Real-time Monitoring Workbench.
Nom de la vue	Nom de la vue à utiliser pour le scénario. Cet attribut est disponible uniquement si l'option **Vue existante** est cochée. Cliquez sur **Parcourir** pour sélectionner la vue.

© Copyright IBM Corp. 2007, 2014

Création de scénarios

Vous pouvez créer un scénario.

Vous pouvez également supprimer des scénarios.

Avertissement : Supprimer un scénario supprime les règles, les alertes et les tableaux récapitulatifs qu'il contient. Une fois supprimés, les règles, alertes et tableaux récapitulatifs ne peuvent pas être restaurés.

Avant de commencer

Pour créer un scénario, vous devez disposer des autorisations suivantes :

- Autorisation de création (voir Chapitre 23, «Autorisations», à la page 295) sur les activités métier (voir «Autorisation de création d'objets», à la page 299 pour plus de détails).
- Autorisation de lecture et écriture (voir Chapitre 23, «Autorisations», à la page 295) sur l'activité métier qui contiendra le scénario.
- (Facultatif) Autorisation de lecture seule sur la vue par défaut du scénario.

Procédure

1. Ouvrez le **Modélisateur de scénarios**.
2. Sélectionnez l'**activité métier** qui contiendra le scénario.
3. Cliquez sur **Nouveau scénario**.
4. Complétez les zones dans la fenêtre Créer un scénario. Pour supprimer un scénario, sélectionnez-le et cliquez sur **Supprimer les scénarios**.

Chapitre 32. Instructions SELECT

Les instructions C-SQL SELECT définissent les vues qui gèrent les informations dans IBM Cognos Real-time Monitoring.

IBM Cognos Real-time Monitoring Workbench crée des instructions SELECT en fonction des vues que vous définissez dans l'interface utilisateur graphique, puis les transmet au(x) serveur(s) Real-time Monitoring en vue de leur instanciation.

Pour visualiser l'instruction SELECT complète qui définit une vue dans Cognos Real-time Monitoring Workbench, cliquez sur **Afficher l'expression SQL** lors de la création ou de la modification d'une vue.

Certaines fonctions avancées de la commande SELECT ne peuvent pas être exprimées par les options de Real-time Monitoring Workbench. Par exemple, vous devez saisir les conditions de jointure complexes, les fenêtres de requêtes et les expressions de table dans l'interface utilisateur.

Cette section décrit en détail la syntaxe et les fonctions de la commande C-SQL SELECT.

La commande C-SQL SELECT est un sous-ensemble et une extension d'ANSI SQL-99, standard de langage de requête. L'implémentation de C-SQL prend en charge les jointures externes à l'aide de la syntaxe de jointure externe ANSI (jointure externe gauche et jointure externe droite) et les fonctions d'agrégation de la clause SELECT. Chaque instruction individuelle est traitée comme une transaction et validée dès son exécution.

Syntaxe

Les spécifications SELECT sont définies à l'aide de plusieurs opérateurs.

Ces opérateurs sont les suivants :

```
SELECT selectList
     FROM joinClauses
     [ WHERE searchCondition ]
     [ GROUP BY groupClause  ]
     [ WINDOW windowClause   ]
     [ ORDER BY orderClause  ]
```

Les opérateurs sont appliqués dans l'ordre suivant :

1. La «Clause FROM», à la page 340 spécifie les tables ou les vues de base qui fournissent des données à cette vue.
2. La «Clause WHERE», à la page 343 (facultative) filtre l'entrée pour qu'elle corresponde aux critères spécifiés.
3. La «Clause GROUP BY», à la page 344 (facultative) regroupe la table finale sur une ou plusieurs colonnes.
4. La «Liste Select», à la page 338 définit les colonnes qui s'afficheront dans la table finale.

© Copyright IBM Corp. 2007, 2014

5. Chapitre 38, «Fenêtres de requêtes», à la page 391 (facultatives) définissent les fenêtres utilisées par les fonctions agrégées de la liste Select.
6. «Sémantique des vues avec état», à la page 346 ORDER BY (facultative) classe (trie) la table résultante.

Liste Select

L'instruction Select list définit les colonnes qui s'affichent dans la table virtuelle finale.

La liste Select suit immédiatement le mot clé SELECT et existe sous les deux formes ci-dessous :

- Un astérisque (*) pour vous permettre de choisir toutes les colonnes qui font partie de la table source.

  ```
  SELECT * ...
  ```

  La vue finale contient les colonnes de chacune des tables ou des vues d'entrée en suivant leur ordre d'affichage dans la source, dans l'ordre indiqué dans la Clause FROM (voir «Clause FROM», à la page 340). Dans le cas de jointures externes, (voir «Jointures externes», à la page 341), les valeurs NULL sont attribuées aux colonnes finales qui n'existent pas dans les deux références.
- Une liste de noms de colonnes uniques ou de colonnes dérivées.

  ```
  SELECT
  columnNameList
  ```

  Où chaque nom columnName est séparé d'un autre nom par une virgule (,) et défini comme

  ```
  columnName[[AS] aliasName [ OVER (windowClause) ] ]
  ```

  L'option AS attribue un nouveau nom à la colonne associée. Le littéral "AS" est facultatif.

  L'option OVER définit une fenêtre en ligne. Pour plus de détails, voir Chapitre 38, «Fenêtres de requêtes», à la page 391.

  Où chaque nom columnName possède l'une des sources suivantes :

  – référence de colonne simple, c'est-à-dire le nom de la colonne.

    ```
    current_rental_price
    ```

  – référence de colonne qualifiée, c'est-à-dire avec une préface correspondant au nom de la table.

    ```
    movie_titles.current_rental_price
    ```

    Pour sélectionner toutes les colonnes d'une table tout en sélectionnant certaines colonnes d'autres tables, utilisez une référence de colonne qualifiée et indiquez un astérisque (*) comme nom de colonne. Par exemple, la syntaxe suivante sélectionne toutes les colonnes movie_title et deux colonnes de la table media :

    ```
    movie_titles.*, media.media_type, media.name
    ```

  – colonne dérivée, c'est-à-dire une expression (ce peut être une expression CASE ; pour plus de détails, voir «Expression CASE», à la page 339) :

    ```
    MAX((movie_titles.current_rental_price/2))
    AS Half_Price
    ```

    Le nom aliasName est requis pour les colonnes dérivées.

Expression CASE

Une expression CASE renvoie le résultat d'une expression correspondant à une condition vraie. Chaque condition peut également renvoyer une valeur NULL.

Si aucune condition vraie n'est trouvée, l'expression renvoie le résultat de la condition ELSE ou NULL si la condition ELSE est omise.

Il existe deux formes d'expressions CASE :

- Condition simple

  Evalue l'expression CASE et la compare au résultat de chaque expression equalsExpression jusqu'à ce qu'elle trouve une correspondance, puis renvoie le résultat resultExpression correspondant. Chaque expression equalsExpression doit être d'un type comparable à l'expression caseExpression.

  ```
  CASE
  caseExpression
      [ { WHEN equalsExpression THEN { resultExpression
  | NULL }}...]
      [ ELSE { resultExpression | NULL } ]
  END
  ```

- Condition de recherche

  Evalue chaque condition searchCondition jusqu'à ce qu'elle trouve une condition vraie, puis renvoie le résultat resultExpression correspondant.

  ```
  CASE WHEN
  searchExpression THEN { resultExpression | NULL }
      [ { WHEN searchExpression THEN { resultExpression
  | NULL }}...]
      [ ELSE { resultExpression | NULL } ]
  END
  ```

  Les expressions CASE suivantes ont le même résultat :

  ```
  SELECT Tier AS
      CASE WHEN Tier = 'High' THEN 'Priority customer'
      ...
  SELECT Tier AS
      CASE Tier WHEN 'High' THEN 'Priority customer'
      ...
  ```

  L'exemple suivant permet de générer des calculs en cours pour chaque état de ticket à chaque niveau :

  ```
  CREATE VIEW VTotal_Tickets AS
      SELECT Tier,
         SUM(CASE Status WHEN Open     THEN 1 ELSE 0 END)
  AS Opens,
         SUM(CASE Status WHEN Reopen   THEN 1 ELSE 0 END)
  AS Reopens,
         SUM(CASE Status WHEN Resolved THEN 1 ELSE 0 END)
  AS Closes,
            ( Opens+Reopens-Closes) AS Pending
         FROM VCustomerTickets
         GROUP BY Tier
  ```

  Le tableau ci-dessous affiche les résultats possibles :

Tableau 93. Résultats de l'expression CASE

Niveau	Ouvertures	Réouvertures	Fermetures	Attente
Elevé	2	0	0	2
Moyen	1	2	3	0
Faible	1	1	1	1

Clause FROM

La clause FROM spécifie les tables et les vues à partir desquelles la nouvelle vue est construite.

```
FROM reference [[ AS ] aliasName ]
  [, reference [[ AS ] aliasName ] ... ]
```

Où une référence est :

- une référence simple :

  ```
  tableOrView
  ```

- une opération de jointure (pour plus de détails, voir «Opérations de jointure») :

  ```
  ( reference [ {LEFT | RIGHT} [ OUTER
  ] | INNER ]
      JOIN reference ON searchCondition )
  ```

- une expression de table (pour plus de détails, voir «Expressions de table», à la page 342) :

  ```
  (
  SELECT selectList FROM joinClauses [ WHERE
  searchCondition ]
       [ GROUP BY groupClause ] [ WINDOW windowClause ])
  ```

La spécification d'une référence unique, simple permet de créer une vue qui est une vue instantanée de la table ou de la vue source. L'insertion de plusieurs références indique une opération de *jointure*.

Contraintes sur les vues

Les vues possèdent des contraintes des sources à la clause FROM.

Ces contraintes sont les suivantes :

- Une vue peut être dérivée d'une table d'événements ou d'une autre vue.
- Une vue peut joindre une table de flux de données ou une vue et une ou plusieurs tables de correspondance.
- Une vue ne peut pas joindre deux ou plusieurs tables de flux de données ou des vues basées sur différentes tables de flux de données. (Un flux de données consolidé est une jointure spéciale de flux d'événements.)
- Une vue ne peut pas être dérivée de tables de correspondance uniquement.

Opérations de jointure

La commande C-SQL SELECT prend en charge plusieurs opérations de jointure.

Ces opérations sont les suivantes :

- «Jointures croisées», à la page 341
- «Jointures internes», à la page 341

- «Jointures externes»
- «Jointures imbriquées», à la page 342

L'insertion de plusieurs références provoque une opération de jointure. La vue finale est une jointure croisée à moins que vous n'utilisiez l'opérateur JOIN pour spécifier un autre type de condition de jointure ou n'insériez une clause WHERE (voir «Clause WHERE», à la page 343) qui spécifie une condition de jointure.

Jointures croisées

Si vous omettez l'opérateur JOIN, vous définissez une vue qui est une jointure croisée des vues d'entrée (également appelée produit cartésien).

L'exemple ci-dessous illustre une jointure des vues Product et Manufacturer :

```
FROM Product AS P, Manufacturer AS M
```

Ne créez une jointure croisée que si vous êtes sûr de le vouloir. Une jointure croisée crée une vue dont le nombre de lignes équivaut au nombre de lignes de la première vue multiplié par le nombre de lignes de la seconde vue (jointure des lignes = lignes vue1 * lignes vue2). Cette jointure affecte considérablement le système et ne produit en général pas la vue que vous voulez.

Au lieu de créer une jointure croisée, indiquez un autre type de jointure avec la clause JOIN :

```
reference [ joinType ] JOIN reference ON searchCondition
```

Jointures internes

La clause JOIN exécute une jointure interne à moins que vous ne spécifiiez une variable *joinType*. Une jointure interne est une jointure dont les lignes de la table de résultat sont en fait les lignes de la première table répondant aux critères spécifiés et associées aux lignes correspondantes de la deuxième table répondant aux critères spécifiés.

```
FROM (Product AS P INNER JOIN Manufacturer
AS M
ON P.productName = M.ProductName)
```

Les jointures internes sont parfois appelées des equijoins.

Jointures externes

Une jointure externe est une jointure dont les lignes de la table résultante sont les lignes qui résultent d'une jointure interne, plus celles de la première table (LEFT OUTER JOIN) ou de la seconde table (RIGHT OUTER JOIN) qui n'ont aucune correspondance dans l'autre table.

Par exemple :

```
FROM (Product AS P LEFT OUTER JOIN Manufacturer
AS M
ON P.productName = M.ProductName
```

La première table d'une LEFT OUTER JOIN (jointure externe gauche) et la seconde table d'une RIGHT OUTER JOIN (jointure externe droite) doivent être une table d'événements ou une vue. Il ne peut s'agir d'une table de correspondance.

Les valeurs NULL sont attribuées aux colonnes finales qui n'existent pas dans les deux références.

Jointures imbriquées

Les jointures peuvent être imbriquées. Il n'existe pas de limite au niveau maximal d'imbrication.

Par exemple :

```
FROM (Product AS P LEFT OUTER JOIN Manufacturer AS M
      ON P.productName = M.ProductName)
      AS Temp, inventoryContext AS INVvt
```

Expressions de table

Une expression de table, également appelée vue en ligne, est une sous-requête qui crée une vue pouvant être référencée par la requête. Il s'agit essentiellement d'une instruction SELECT, insérée entre parenthèses, qui s'affiche dans la clause FROM.

Par exemple, la syntaxe suivante est une expression de table contenue dans une requête :

```
SELECT *
    FROM Warehouse AS wh,
(SELECT *
            FROM WarehouseQtyChange AS wqc,
                 Product AS pr
            WHERE wqc.wprod_id = pr.pprod_id)
AS sv,
    WHERE sv.warehouse_id = wh.wh_region_id
```

L'exemple ci-dessus a le même résultat que `WHRegionView` dans cet exemple :

```
SummaryView:
SELECT *
    FROM WarehouseQtyChange AS wqc,
         Product AS pr
    WHERE wqc.wprod_id = pr.pprod_id
WHRegionView:
SELECT *
    FROM Warehouse AS wh,
SummaryView AS sv
    WHERE sv.warehouse_id = wh.wh_region_id
```

Syntaxe

Une expression de table est une instruction SELECT limitée, entre parenthèses, à laquelle s'appliquent des restrictions.

```
( SELECT selectList FROM joinClauses
     [ WHERE searchCondition ]
     [ GROUP BY groupClause ]
     [ WINDOW windowClause ]  ) AS aliasName
```

Restrictions

Les mêmes restrictions sémantiques s'appliquent aux vues en ligne et aux vues standard.

Par exemple, une colonne dérivée d'une vue en ligne ne peut pas avoir le même nom ou le même alias qu'une colonne de la liste selectList. Des restrictions et des limitations leur sont appliquées.

Ces restrictions et limitations sont les suivantes :

- Elles doivent être entre parenthèses.
- Elles doivent être dans le même flux de données de base que les autres vues et tables de la requête.
- Elles doivent êtres définies uniquement dans la clause FROM.

Exemple de clause HAVING

Certaines implémentations SQL incluent une clause HAVING qui vous permet de filtrer les résultats agrégés d'une vue. C-SQL n'inclut pas la clause HAVING. Cependant, vous pouvez créer une clause HAVING à l'aide d'une clause WHERE.

Une clause WHERE (voir «Clause WHERE») filtre les résultats d'une vue en ligne. Essentiellement :

```
SELECT * FROM (inlineView) WHERE filterCondition
```

Par exemple, pour créer une vue qui filtre le résultat d'une agrégation, vous devez d'abord exécuter l'agrégation dans une vue en ligne, puis filtrer les résultats dans la vue finale. La vue en ligne suivante additionne les ventes totales pour chaque gamme de produits, puis la vue finale affiche, dans l'ordre décroissant, uniquement les résultats supérieurs à 1 000 000 $.

```
SELECT FamilyTotals.family AS "Product Line",
       FamilyTotals.sales_for_family AS "Total Sales"
    FROM (SELECT family,
                 SUM(total_price) AS sales_for_family
              FROM OrdQtyDemand
              GROUP BY Family
          ) AS FamilyTotals
    WHERE "Total Sales" > 1000000
    ORDER BY "Total Sales" DESC;
```

Dans l'exemple ci-dessus, lorsque de nouveaux événements sont entrés dans la vue OrdQtyDemand, les totaux sont mis à jour et l'ordre des gammes de produits peut changer. De fait, de nouvelles gammes peuvent être entrées dans la vue lorsque leurs ventes totales excèdent un million.

Clause WHERE

Une clause WHERE examine chaque ligne entrée et accepte uniquement celles qui correspondent à la condition spécifiée.

La syntaxe est la suivante :

```
WHERE searchCondition
```

Une condition *searchCondition* est une combinaison de prédicats booléens qui effectuent ensemble un test. Seules les lignes en entrée qui passent le test avec succès sont insérées dans la nouvelle vue. Les lignes qui ne remplissent pas la condition sont supprimées, et ne sont ni suivies ni incluses dans les calculs d'une fonction d'ensembles (Set), d'une fonction d'ensembles mobiles (Moving set) ou d'une fonction de classement (Rank).

Toutes les vues dépendantes sont mises à jour et leurs fonctions recalculées que les lignes entrées remplissent ou non la condition. Pour plus d'informations, voir «Mise à jour des vues par le biais de la propagation d'événements», à la page 370.

Prédicats

Un prédicat est une expression booléenne qui déclare un fait sur des valeurs. Chaque expression peut être déclarée seule ou comparée à l'une des valeurs de test booléennes.

Par exemple, ces expressions sont équivalentes :

```
WHERE (Age >= 21)
WHERE (Age >= 21) IS TRUE
```

Les prédicats pris en charge par la clause WHERE sont répertoriés dans la section Chapitre 22, «Opérateurs et constantes», à la page 291.

Les prédicats peuvent inclure des fonctions, mais les fonctions qui renvoient à des colonnes ne peuvent renvoyer qu'à des colonnes d'événements.

Alias

Le langage C-SQL étend la norme ANSI pour autoriser les références aux alias dans la clause WHERE.

Par exemple :

```
SELECT user_age AS Age
   FROM user_list
   WHERE (Age >= 21)
```

Clause GROUP BY

La clause GROUP BY regroupe la table virtuelle finale sur une ou plusieurs colonnes.

La syntaxe est la suivante :

```
GROUP BY [tableName.]columnName[ , [tableName.]columnName
... ]
```

Où *tableName* est une table ou une vue source. Toutes les variables *columnName* de la liste de sélection qui ne sont pas désignés directement par une fonction de classement (Rank) ou une fonction scalaire (Scalar) peuvent s'afficher dans la liste GROUP BY.

L'autre méthode plus performante permettant de regrouper des données consiste à utiliser la clause PARTITION des fenêtres. Pour plus de détails, voir «Partitions des fenêtres», à la page 401.

L'option GROUP BY fournit un récapitulatif d'informations des groupes de lignes dont les valeurs sont identiques à celles des zones sélectionnées. Regardez cet ensemble de données :

```
Name          Quantity
------------  --------
Nano Webber         10
Fizzy Lifter       700
```

```
Nano Webber         50
Nano Webber         20
Nano Webber         15
Smoke Shifter      310
```

Si vous créez une vue qui regroupe des valeurs par nom et détermine la somme de la quantité de chaque groupe, cette vue ressemblera à ceci :

```
SELECT product.name AS Name, SUM(product.quantity) AS
Qsum
    FROM product
    GROUP BY product.name
Name          Qsum
------------  ----
Nano Webber     95
Fizzy Lifter   700
Smoke Shifter  310
```

Vous pouvez aussi procéder à des regroupements sur plusieurs zones. Par exemple :

```
SELECT product.name AS Name, product.location AS Locale,
        SUM(product.quantity) AS Qsum
    FROM product
    GROUP BY product.name, product.location
Name          Location Qsum
------------  -------- ----
Nano Webber   West       10
Fizzy Lifter  East      700
Nano Webber   East       85
Smoke Shifter West      310
```

Si la liste de sélection inclut une fonction d'ensembles mobiles (Moving set), chaque groupe contient un résultat correspondant à cette fonction.

Alias

Le langage C-SQL étend la norme ANSI pour autoriser les références aux alias dans la clause GROUP BY.

Par exemple :

```
SELECT product.name AS Name,product.location
AS Locale,
        SUM(product.quantity) AS Qsum
    FROM product
GROUP BY Name, Locale
```

Vues dérivées

Si une vue est définie avec une clause GROUP BY, toutes les vues dérivées de cette vue ont une clause GROUP BY implicite. On parle alors de fusion ou de développement de vues.

Ainsi, dans l'illustration suivante, même si l'instruction SELECT de la Vue2 n'inclut pas de clause GROUP BY, ses résultats incluent les mêmes groupes que la Vue1.

Dans l'illustration ci-dessous, les tables Flux de données et Table de correspondance sont utilisées dans l'instruction SELECT pour créer la Vue1.

```
SELECT Name,Qty, Cost FROM Data_Stream, Prices GROUP
BY Name WHERE Data_Stream.Name=Lookup_Table.Name
```

La Vue2 est ensuite créée à partir de la Vue1 à l'aide de l'instruction suivante :

```
SELECT Name,(Qty*Cost)AS Total from View1
```

Figure 37. Exemple de vue dérivée

Sémantique des vues avec état

Si une vue contient une clause GROUP BY, cette vue est une vue avec état. Elle conserve les informations des événements précédents et pas seulement celles du dernier événement.

Ainsi, la vue View2, de l'exemple ci-dessus «Vues dérivées», à la page 345, est une vue avec état même si sa définition SELECT ne comprend pas de fonction d'ensembles ou de clause GROUP BY explicite, car elle est dérivée d'une vue avec état.

Pour en savoir davantage, voir «Vues sans état et vues avec état», à la page 370.

Clause ORDER BY

La clause ORDER BY classe (trie) la vue finale en fonction des noms de colonnes ou des résultats d'expression.

La syntaxe est la suivante :

```
ORDER BY columnName [{ASC|DESC}] [ , [columnName [{ASC|DESC}]
] ...
```

Sans cette clause, il n'y a aucune garantie que la même requête produira des lignes respectant le même ordre dans les requêtes suivantes.

Toutes les clés de tri mentionnées dans la clause ORDER BY doivent désigner un nom de colonne de la liste de sélection.

Par défaut, la vue est classée par ordre croissant (ASC). Pour classer par ordre décroissant, spécifiez l'option DESC.

L'exemple suivant trie la vue d'abord par nom de fournisseur dans l'ordre croissant, puis par prix, pour chaque fournisseur, dans l'ordre décroissant :

```
SELECT Product.prod_id AS ProductID,
        orderStatusDataStream.OS_PRICE AS Price,
```

```
        Supplier.supp_name AS SupplierName
    FROM orderStatusDataStream, Product, Supplier
    WHERE orderStatusDataStream.OS_PROD_ID = Product.prod_id
AND
        Product.prod_supp_id = Supplier.supp_id
ORDER BY Supplier.supp_name
ASC, orderStatusDataStream.OS_PRICE DESC
```

Chapitre 33. TIBCO Rendezvous

TIBCO Rendezvous est un système de messagerie destiné aux applications métier.

Ces applications publient des messages vers le flux géré par les serveurs de transport TIBCO Rendezvous. Chaque message porte un nom qui identifie son objet.

Les applications contrôlent le flux de recherche des messages qui, lorsqu'ils sont trouvés, sont fournis à d'autres applications, telles que les agents TIBCO Rendezvous d'IBM Cognos Real-time Monitoring (voir «Agents TIBCO Rendezvous», à la page 40).

Fonctionnement de TIBCO Rendezvous

Les tables de flux de données IBM Cognos Real-time Monitoring reçoivent les messages TIBCO Rendezvous sous la forme d'événements. Chaque table de flux de données correspond à un seul objet de message. Les tables identifient les objets des messages auprès d'une application démon d'écoute par le biais de l'agent. Lorsque le démon localise un nouveau message avec l'objet demandé, il transmet le message à la table via l'agent. La définition de la table fait ensuite correspondre ce message à la table en tant que nouvel événement.

Figure 38. Fonctionnement de TIBCO Rendezvous

Tables TIBCO Rendezvous

Une table de flux de données TIBCO Rendezvous reçoit les messages d'une application métier par le biais d'un flux de messages TIBCO Rendezvous.

Chaque message est identifié par son objet, et chaque nouveau message correspondant à un objet est un nouvel événement. Lorsque la table reçoit un nouveau message, elle fait d'abord correspondre les données du message aux types de données qu'elle contient.

Limitations

Tous les messages correspondant à un objet d'événement doivent avoir la même forme : ils doivent disposer des mêmes zones, même si une zone peut être vide.

© Copyright IBM Corp. 2007, 2014

Certains types de données TIBCO Rendezvous ne sont pas pris en charge et ne peuvent pas correspondre à une table de flux de données.

Pour en savoir davantage, voir «Types de données TIBCO Rendezvous», à la page 354.

Création d'une table de flux de données TIBCO Rendezvous

Utilisez la procédure suivante pour créer une table de flux de données TIBCO Rendezvous.

Pour en savoir davantage, voir «Informations sur les colonnes TIBCO», à la page 352.

Avant de commencer

Pour créer un flux de données TIBCO Rendezvous, vous devez disposer des éléments suivants :

- Des autorisations

  Autorisation de création (voir Chapitre 23, «Autorisations», à la page 295) sur les tables (voir «Autorisation de création d'objets», à la page 299) et autorisation Lecture seule sur l'agent qui alimente la table.
- Un agent

  Un agent TIBCO Rendezvous existant qui se connecte aux flux de messages TIBCO Rendezvous. Créez un agent dans l'onglet d'administration du plan de travail. Pour plus de détails, voir «Agents TIBCO Rendezvous», à la page 40.
- Un nom d'objet

  Chaque message TIBCO Rendezvous comporte un nom d'objet qui identifie la source de l'événement. Vous identifiez le nom d'objet, et l'agent contrôle le flux de messages pour y rechercher les messages. Lorsqu'il en trouve un, il transmet les informations afférentes à la table de flux de données. Les noms d'objet sont constitués d'un ou de plusieurs éléments séparés par des points, par exemple : SUPPORT.TICKETS.
- Un format de message

  Chaque message TIBCO Rendezvous contient des zones d'informations. Ces informations peuvent elles-mêmes comporter des messages imbriqués, un objet, et des en-têtes "reply to". Pour en savoir davantage, voir «Informations sur les colonnes TIBCO», à la page 352.
- Un fichier exemple (facultatif)

  Si le message tibco_complex_string comporte une chaîne complexe, il est utile de disposer d'un exemple de fichier qui contient des données au format de la chaîne d'événement réelle. Vous pouvez utiliser cet exemple lorsque vous créez le flux de données afin de vérifier que les zones sont correctement mappées vers la table de flux de données. Pour ce faire, examinez la manière dont les données sont alignées dans les colonnes.

Pour de plus amples informations sur le nom d'objet et le format de message, consultez l'informaticien qui gère votre système TIBCO Rendezvous. Une table de flux de données TIBCO Rendezvous possède les attributs décrits dans le tableau ci-dessous :

Tableau 94. Attributs de la table de flux de données de TIBCO Rendezvous

Attribut	Description
Nom	Identifie l'objet de flux de données. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Enregistrer dans	Indique le dossier dans lequel l'objet de flux de données sera stocké. Le bouton **Sélectionner un dossier** permet de sélectionner ou de créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Description	Description facultative qui peut contenir tous les caractères de texte.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Si les événements peuvent être traités dans n'importe quel ordre, désactivez cet attribut. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Agent TIBCO Rendezvous	Agent existant qui se connecte au flux de messages TIBCO Rendezvous. Créez un agent depuis l'onglet d'administration du plan de travail. Pour plus de détails, voir «Agents TIBCO Rendezvous», à la page 40.
Abonnement	Identifie l'objet sous lequel le message est envoyé et défini par son éditeur. En général, cette chaîne ressemble à celle-ci : com.cognos.obi.mytibcotopic.
Informations sur la colonne	Les zones **Informations sur la colonne** déterminent comment placer les zones du message TIBCO Rendezvous dans les colonnes de la table de flux de données. Cette dernière contient une colonne pour chaque zone. Pour en savoir davantage, voir «Informations sur les colonnes TIBCO», à la page 352.
Intervalle de suppression de l'état	Cet onglet contient plusieurs options permettant de supprimer les données d'événements sauvegardées. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Flux de données**.
4. Choisissez le type de flux de données TIBCO RV.
5. Identifiez l'objet des événements de message à collecter.
6. Sélectionnez un agent TIBCO Rendezvous existant.
7. Identifiez les zones du message et déterminez comment elles correspondent aux types de données IBM Cognos Real-time Monitoring.
8. Eventuellement, définissez l'intervalle de suppression de l'état dans l'onglet correspondant.
9. Enregistrez la table TIBCO Rendezvous.

Informations sur les colonnes TIBCO

Les zones Informations sur la colonne déterminent comment placer les zones du message TIBCO Rendezvous dans les colonnes de la table de flux de données.

Cette dernière contient une colonne pour chaque zone.

Chaque zone du message peut être une zone simple qui s'insère directement dans une colonne de la table de flux de données ou une zone complexe (zone de fichier à plat) constituée de plusieurs zones dont chacune s'insère dans des colonnes de la table. Les zones complexes sont considérées comme des fichiers à plat aux formats délimité (CSV), largeur fixe ou XML. Pour obtenir la description détaillée de ces types de fichier, voir Chapitre 13, «Fichiers à plat», à la page 103.

Chaque colonne de la table de flux de données possède les attributs décrits dans le tableau ci-dessous.

Tableau 95. Attributs des colonnes TIBCO Rendezvous

Attribut	Description
Nom de zone	Nom de la colonne dans la table de flux de données.
Nom du message	Nom de la zone dans le message. Lorsque vous associez une zone de message, le nom de chaque zone incorporée est Non disponible et il n'est pas modifiable.
Type de données	Type de données associé à la colonne. Pour en savoir davantage, voir «Types de données TIBCO Rendezvous», à la page 354.
Format	(Facultatif) Format de la colonne d'événement des valeurs VARCHAR (chaîne) et DECIMAL.

Vous ajoutez des colonnes en cliquant sur l'option **Ajouter une zone** ou **Ajouter une zone Fichier à plat**.

Vous pouvez ajouter des colonnes pour l'objet et la réponse du message. Pour mapper l'objet vers une colonne de la table de flux de données, indiquez <tibcosubject> comme nom de zone. Pour les réponses, indiquez <tibcoreply>.

Si une zone appartient à un message imbriqué, vous pouvez définir la zone à associer dans la colonne de la table de flux de données à l'aide d'une notation

décimale. Par exemple, si vous souhaitez mapper une zone d provenant d'un message c qui est imbriqué à un message b, utilisez le format de nom de zone <a.b.c.d.

Ajout d'une zone de message

Chaque zone du message peut être une zone simple qui s'insère directement dans une colonne de la table de flux de données ou une zone complexe (zone de fichier à plat) constituée de plusieurs zones dont chacune s'insère dans des colonnes de la table.

Les zones complexes sont traitées en tant que fichiers à plat au format CSV, à largeur fixe ou XML. Pour obtenir la description détaillée de ces types de fichier, voir Chapitre 13, «Fichiers à plat», à la page 103.

Procédure

1. Cliquez sur **Ajouter une zone Fichier à plat**.

   Une zone fichier à plat crée, dans le message, une zone contenant plusieurs zones qui s'insèrent chacune dans une colonne de la table de flux de données.
2. Choisissez le type Fichier à plat de la zone de message.

   Identifiez un fichier exemple qui va vous aider à mapper les colonnes. Ce fichier est un échantillon du fichier de données réel. Les données de ce fichier s'affichent à l'étape suivante pour vous aider à faire correspondre les données de flux de données à la table. Ceci est facultatif pour les fichiers délimités et à largeur fixe.
3. Pour les fichiers à largeur fixe, définissez l'emplacement des colonnes de données dans la fenêtre Définir les largeurs des zones.
4. Identifiez les attributs de fichier à plat. Pour plus de détails, voir «Tables de flux de données associées aux fichiers à plat», à la page 104.
5. Définissez les informations sur la colonne en fonction du format choisi. Pour plus de détails sur le type de source, voir les rubriques suivantes :
   - «Fichiers à largeur fixe», à la page 110
   - «Fichiers délimités», à la page 110
   - «Prise en charge des fichiers XML», à la page 111
6. Cliquez sur **Enregistrer le flux de données** pour enregistrer la définition de la zone de message.

Modification de la définition d'une zone de message

Chaque zone du message peut être une zone simple qui s'insère directement dans une colonne de la table de flux de données ou une zone complexe (zone de fichier à plat) constituée de plusieurs zones dont chacune s'insère dans des colonnes de la table.

Les zones complexes sont considérées comme des fichiers à plat aux formats délimité (CSV), largeur fixe ou XML. Pour obtenir la description détaillée de ces types de fichier, voir Chapitre 13, «Fichiers à plat», à la page 103.

Procédure

Dans la colonne **Format** de la zone, remplacez **Fichier à plat : type de fichier** par **<Changer la mise en forme>**.
Lors de la modification d'une zone de message, l'option de fichier exemple pour les types de fichiers délimités et à largeur fixe n'est pas disponible.

Types de données TIBCO Rendezvous

Chaque message est un ensemble de zones qui contiennent chacun une donnée d'un type spécifique. Vous pouvez identifier chaque zone à l'aide de son nom défini et spécifier le type de données associé à la zone.

Les types de données TIBCO Rendezvous correspondent aux types de données Cognos Real-time Monitoring (voir Chapitre 9, «Types de données», à la page 69) comme indiqué dans le tableau ci-dessous :

Tableau 96. Mappage des types de données TIBCO Rendezvous

Type TIBCO Rendezvous	Type Cognos Real-time Monitoring
Types de données personnalisés	Pas de prise en charge
BOOL	Booléen
DATETIME	Horodatage
F32	Double
F32ARRAY	Pas de prise en charge
F64	Double
F64ARRAY	Pas de prise en charge
I16	Entier
I16ARRAY	Pas de prise en charge
I32	Entier
I32ARRAY	Pas de prise en charge
I64	Long
I64ARRAY	Pas de prise en charge
I8	Entier
I8ARRAY	Pas de prise en charge
IPADDR32	Pas de prise en charge
IPPORT16	Pas de prise en charge
MSG	Pas de prise en charge
OPAQUE	Pas de prise en charge
STRING	Varchar
U16	Entier
U16ARRAY	Pas de prise en charge
U32	Long
U32ARRAY	Pas de prise en charge
U64	Décimal
U64ARRAY	Pas de prise en charge
U8	Entier
U8ARRAY	Pas de prise en charge
XML	Varchar

Chapitre 34. Utilisateurs

Chaque utilisateur qui interagit avec IBM Cognos Real-time Monitoring est connu du système grâce aux informations de son compte utilisateur.

Lorsque des administrateurs créent ou modifient des comptes ou lorsque des utilisateurs affichent leurs paramètres de compte, ils accèdent aux informations de compte spécifiques à partir de ces onglets.

Utilisateur rtmadmin

Chaque installation comporte un utilisateur par défaut appelé rtmadmin. Cet utilisateur dispose de toutes les autorisations. Il crée d'autres utilisateurs et leur attribue des autorisations. La possibilité de créer des utilisateurs figure au nombre des autorisations que l'utilisateur rtmadmin peut accorder. Contactez votre administrateur pour prendre connaissance du mot de passe de l'utilisateur rtmadmin par défaut de votre installation.

Modification de vos propres informations de compte

Il vous est possible de modifier vos propres informations de compte.

Procédure

Cliquez sur **Paramètres du compte** à tout moment.

Création, suppression ou modification d'un compte utilisateur

Vous pouvez créer, supprimer ou modifier un compte utilisateur.

Vous ne pouvez pas supprimer l'utilisateur système ni vous supprimer.

Procédure

1. Ouvrez la **Console d'administration**.
2. Cliquez sur **Utilisateurs** pour accéder à la liste de tous les utilisateurs actuellement définis dans le système. Dans cette page, vous pouvez effectuer les actions suivantes :
   - Modifier les informations d'un compte utilisateur existant en cliquant deux fois sur le nom dans la liste.

Onglet Détails de l'utilisateur

L'onglet **Détails de l'utilisateur** répertorie les attributs utilisateur.

Tableau 97. Attributs de l'onglet Détails de l'utilisateur

Attribut	Description
Nom d'utilisateur	Nom de connexion de l'utilisateur. Ce nom doit être unique parmi tous les objets du dossier racine, les activités métier, les utilisateurs, les rôles, les objets de tableau de bord, etc.
Mot de passe	Mot de passe de l'utilisateur. Le mot de passe peut contenir toute les combinaisons de lettres, de chiffres et de caractères.

© Copyright IBM Corp. 2007, 2014

Tableau 97. Attributs de l'onglet Détails de l'utilisateur (suite)

Attribut	Description
Rôles	Rôles auxquels cet utilisateur appartient.

Onglet Profils de diffusion

Les profils de diffusion indiquent où et comment envoyer des alertes et des données à l'utilisateur.

Les utilisateurs peuvent posséder plusieurs profils. Dans le **Gestionnaire d'alertes** d'IBM Cognos Real-time Monitoring Dashboard, les utilisateurs peuvent identifier quels profils reçoivent quelles alertes. Par ailleurs, tous les utilisateurs disposent au moins d'un profil : **Profil tableau de bord**. Le **Profil tableau de bord** envoie des notifications au **Gestionnaire d'alertes** de Real-time Monitoring Dashboard.

L'un des profils au moins doit être désigné comme celui devant être utilisé lors de l'abonnement à une alerte. Tous les profils marqués **Auto** sont automatiquement ajoutés aux nouveaux abonnements d'alertes.

Les types de profils correspondent aux mécanismes de diffusion possibles disponibles dans l'installation. Ils incluent les éléments suivants :

- **Profil tableau de bord**

  Gestionnaire d'alertes dans le tableau de bord. Vous ne pouvez pas supprimer ce profil.
- **Courrier électronique**

  Compte de messagerie électronique permettant de recevoir le message d'alerte.
- **Service Web**

  Méthode de services Web qui reçoit la notification d'alerte et le tableau récapitulatif joint en tant que données XML. Pour plus d'informations, voir Chapitre 37, «Services Web», à la page 377.

Courrier électronique

Il s'agit d'une adresse à laquelle l'utilisateur reçoit des messages électroniques.

Les messages sont pourvus des attributs suivants :

Tableau 98. Attributs de courrier électronique

Attribut	Description
Nom du profil	Indique le nom qui identifie le profil. Ce nom ne peut comporter que des lettres et des chiffres.
Adresse électronique	Indique l'adresse électronique de l'utilisateur.

Service Web

Le service Web reçoit les notifications d'alerte et tous les tableaux récapitulatifs joints en tant que données XML.

Ce service publie un message SOAP (Simple Object Access Protocol) de style doc, et non RPC (Remote Procedure Call).

Le service Web possède les attributs suivants :

Tableau 99. Attributs du service Web

Attribut	Description
Adresse URL du service Web	Emplacement HTTP de l'application fournissant le service DOC (SOAP). Les messages de style RPC ne sont pas pris en charge.
Méthode	Méthode du service Web à utiliser.
Nom d'utilisateur	Nom du compte à utiliser pour se connecter au service (facultatif).
Mot de passe	Mot de passe du compte (facultatif).
UDF	Fonction définie par l'utilisateur (UDF) permettant de lancer le service Web. Si aucune fonction définie par l'utilisateur n'est indiquée, le langage WSDL par défaut est utilisé pour appeler le service Web. Pour plus d'informations, voir Chapitre 35, «Fonctions définies par l'utilisateur», à la page 359 ainsi que les sections sur l'utilisation des fichiers JAR et UDF dans le *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*. (facultatif).

Pour utiliser un service Web, votre administrateur doit désigner le fichier AlertMessage.xsd et le fichier AlertService.wsdl qui décrit le service d'abonnement, les données qu'il fournit et le mode d'échange des données avec le service.

Votre administrateur peut localiser ces fichiers dans le répertoire :

emplacement d'installation`/realtime/sdk/api/wsalert/api/wsalert`

Pour plus d'informations sur le service Web, voir Chapitre 37, «Services Web», à la page 377.

Onglet Droits d'accès

Les droits d'accès déterminent le niveau d'accès d'un utilisateur ou d'un rôle sur un objet. Ils peuvent être attribués à une classe d'objets ou à un objet spécifique.

Dans l'onglet **Droits d'accès**, vous pouvez attribuer des autorisations aux éléments suivants :

- Activités métier
- Vues, cubes et dimensions
- Tables de correspondance et flux de données
- Utilisateurs
- Rôles
- Agents
- Propriétés générales du système
- Tableau de bord
- Objets de tableau de bord
- Processus externes

Les nouveaux utilisateurs et nouveaux rôles ne disposent d'aucun accès sur les objets. Par conséquent, les utilisateurs et les rôles ne peuvent recevoir et voir que les alertes et les tableaux récapitulatifs associés à un abonnement obligatoire.

Pour plus d'informations sur l'attribution des autorisations, voir Chapitre 23, «Autorisations», à la page 295.

Chapitre 35. Fonctions définies par l'utilisateur

Les fonctions définies par l'utilisateur (UDF) vous permettent d'étendre C-SQL en définissant vos propres fonctions à utiliser dans les requêtes, les vues et les règles.

Vous pouvez ainsi définir une fonction scalaire ou une fonction d'ensembles en implémentant les interfaces Java appropriées d'IBM Cognos Real-time Monitoring.

Les fonctions UDF sont des programmes Java qui utilisent des arguments et renvoient une valeur, exactement comme les fonctions internes de Cognos Real-time Monitoring. Par exemple, vous pouvez avoir une UDF qui utilise un ensemble de valeurs et les concatène par ordre alphabétique, tout en ignorant les valeurs NULL. Vous utiliserez cette fonction UDF dans une expression comme celle-ci :

```
ConcatSet(Product.Name)
```

Après avoir compilé le programme Java, vous le déployez (chargez) dans Cognos Real-time Monitoring où il est ensuite à la disposition de tous les utilisateurs qui peuvent créer ou modifier des requêtes, des vues et des règles.

Pour plus d'informations sur la création des fonctions définies par l'utilisateur, voir «Création et utilisation d'une fonction définie par l'utilisateur», à la page 360. Par ailleurs, pour plus de détails sur les interfaces, voir 'com.cognos.obi.api.function' de la documentation Javadoc.

Restrictions applicables aux fonctions définies par l'utilisateur (fonctions UDF)

Plusieurs restrictions s'appliquent aux fonctions définies par l'utilisateur.

Ces restrictions sont les suivantes :

- Il s'agit uniquement des fonctions scalaires (Scalar) et d'ensembles (Set), et non des fonctions de classement (Rank).

  Les fonctions UDF peuvent définir des fonctions scalaires en implémentant l'interface IUDScalarFunction ou des fonctions d'ensembles en implémentant les interfaces IUDAggregateFunction et IUDAggregateState. Pour plus de détails, voir "com.celequest.api.function" de la documentation javadoc.
- Les fonctions UDF ne sont pas exposées dans IBM Cognos Real-time Monitoring Workbench.

  L'éditeur de formules ne reconnaît pas les fonctions UDF. Ainsi, vous êtes chargé d'effectuer le suivi des fonctions UDF enregistrées sur le système.
- Tous les utilisateurs ont accès à toutes les fonctions UDF.

  Vous ne pouvez pas appliquer d'autorisations d'accès sur une fonction UDF. De même, plusieurs utilisateurs peuvent définir différentes fonctions UDF portant un même nom. Le cas échéant, le système utilise la première qu'il trouve.
- Les fonctions d'ensembles doivent implémenter la sérialisation d'objet et gérer la compatibilité ascendante.

© Copyright IBM Corp. 2007, 2014

La non-implémentation des sous-programmes pertinents de sérialisation/désérialisation peut provoquer un comportement imprévisible dans de nombreuses zones, notamment l'exécution du point de contrôle et de la récupération et l'exécution parallèle.

- Les fonctions UDF ne sont jamais envoyées comme prédicats à une source distante.

  Les fonctions UDF ne sont jamais envoyées à un SGBD en vue d'être évaluées. Pour plus de détails, voir «Limitations de la colonne dans les requêtes de tables de correspondance», à la page 268.

Création et utilisation d'une fonction définie par l'utilisateur

Ces instructions utilisent l'exemple de fonction UDF ConcatSet.

Cet exemple se trouve à l'emplacement suivant : `\realtime\webcontent\sdk\udf`. Pour plus de détails sur l'exemple, consultez le fichier `README.txt` de ce répertoire.

Procédure

1. Créez votre fonction UDF en implémentant les interfaces appropriées com.cognos.obi.api.function.

   Pour plus de détails sur les interfaces, consultez la documentation javadoc.
2. Compilez votre implémentation.

   Lorsque vous compilez l'implémentation de la fonction UDF, incluez le fichier cognosAPI.jar. Ce fichier se trouve à l'emplacement suivant : `\realtime\webcontent\sdk\udf`. Pour plus de détails sur l'exemple, consultez le fichier `README.txt` de ce répertoire.
3. Créez un fichier manifeste pour le fichier JAR UDF.

   Un fichier manifeste est un fichier XML qui décrit le fichier JAR UDF à Cognos Real-time Monitoring. Pour obtenir la description du fichier et un exemple, voir «Fichiers manifestes», à la page 361.
4. Créez un fichier JAR contenant vos classes, par exemple ConcatSet.class, et le fichier manifeste (manifest.xml). Le fichier manifeste doit s'afficher sous com/cognos/obi/manifest dans le fichier JAR. Par exemple :

   ```
   jar -cvf udf.jar samples/* com/cognos/obi/manifest/manifest.xml
   ```
5. Ajoutez la fonction UDF à la liste des fonctions UDF disponibles.
   - Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
   - Cliquez sur le dossier **Fonctions définies par l'utilisateur**, et choisissez **Nouvelle fonction définie par l'utilisateur**.
   - Spécifiez ou chargez le fichier JAR qui contient la fonction, puis cliquez sur **Continuer**.
   - Sélectionnez les fonctions à ajouter, puis attribuez-leur éventuellement de nouveaux noms.
   - Cliquez sur **Terminer** pour ajouter les fonctions UDF à la liste.

   Une fois la fonction UDF déployée, vous pouvez l'utiliser dans des formules semblables à celle-ci :

   ```
   ConcatSet(Product.Name)
   ```

Modification d'une fonction définie par l'utilisateur

Vous avez la possibilité de modifier une fonction UDF existante.

Procédure

1. Modifiez l'implémentation et créez un fichier JAR mis à jour.
2. Téléchargez ce fichier JAR pour remplacer le fichier existant.
3. Ajoutez de nouveau les fonctions UDF à la liste des fonctions définies par l'utilisateur.

Fichiers manifestes

Un fichier manifeste est un fichier XML qui décrit le contenu du fichier JAR UDF à IBM Cognos Real-time Monitoring.

Pour toutes les fonctions du fichier JAR, définissez les éléments `<UDF>` et `<name>`. Pour tous les types de données que la fonction peut renvoyer, définissez un élément <UDFDescriptor>. Si des appels ultérieurs à une même fonction possédant les mêmes valeurs d'arguments peuvent renvoyer des valeurs différentes, définissez l'élément `<isVariant>` sur true. Par exemple, la fonction interne CURRENT_TIMESTAMP (voir «CURRENT_TIMESTAMP», à la page 144) n'utilise aucun argument, mais renvoie un résultat différent chaque fois qu'elle est appelée. Cette fonction est de type "variant".

Ci-après figure un exemple de fichier manifeste pour la fonction Concatset(). Cette liste est adaptée du fichier contenu dans le répertoire samples situé dans `/samples/udf/jar/com/cognos/obi/manifest.xml`.

Exemple de fichier manifest.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jarManifest xsi:schemaLocation="http://obi.cognos.com/4
jarManifest.xsd"
xmlns="http://obi.cognos.com/4"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<jarManifestXSDVersion>1</jarManifestXSDVersion>
  <author>by Martin Handwerker &lt;martin@handwerkers.com&gt;</author>
    <UserDefinedFunctions>
      <UDF>
        <name>ConcatSet</name>
        <description>Concatenate a set of values alphabetically
into a large string. Ignores nulls.</description>
        <implementor>com.udfobi.concatset.ConcatSet</implementor>
        <UDFDescriptors>
           <UDFDescriptor>
             <result>VARCHAR</result>
             <argument>
                <type>ANY</type>
             </argument>
           </UDFDescriptor>
        </UDFDescriptors>
        <isVariant>false</isVariant>
      </UDF>
    </UserDefinedFunctions>
</jarManifest>
```

Chapitre 36. Vues

Les vues métier sont des modèles de données qui offrent une image en temps réel d'une activité métier.

Les enregistrements des modifications et des transactions de votre activité métier sont entrés dans IBM Cognos Real-time Monitoring en tant qu'événements. Chaque nouvel événement provoque une mise à jour immédiate des vues (les modèles de gestion) dérivées de cet événement, offrant une image en temps réel des mesures métier. En outre, dès qu'une vue a été mise à jour, le système évalue les règles qui lui sont associées recherchant des conditions métier exceptionnelles requérant de l'attention.

Une vue métier est une table virtuelle qui réside en mémoire et dont le contenu provient de plusieurs sources définies par une instruction de requête C-SQL SELECT. Vous définissez les vues dans IBM Cognos Real-time Monitoring Workbench. Ce système crée ensuite une instruction SELECT bien formée avant de la transmettre aux serveurs Cognos Real-time Monitoring en vue de son exécution et de sa maintenance. Pour plus d'informations sur l'instruction SELECT, sa syntaxe et son utilisation, voir Chapitre 32, «Instructions SELECT», à la page 337.

Vous pouvez limiter les lignes accessibles à un utilisateur en associant un filtre d'accès à la vue et en appliquant ce filtre aux utilisateurs ou aux rôles qui affichent cette vue. Pour plus d'informations, voir Chapitre 2, «Filtres d'accès», à la page 3.

Les données des vues des tables de correspondance sont statiques ou bien changent lentement. Lorsque des modifications sont appliquées dans les tables de correspondance, le moteur de requête ne met pas à jour la vue. En revanche, ces modifications sont reflétées dans la vue lorsque la table de correspondance est jointe à la ligne de la table de flux de données.

Création de vues

Vous pouvez créer des vues qui sont mises à jour par les modifications et les transactions de votre activité commerciale.

Conseil :

Utilisez des vues sans état, si possible. Les vues sans état utilisent beaucoup moins de mémoire que celles avec état, compte tenu que seul le dernier événement est conservé en mémoire. Utilisez les vues sans état pour fournir des données aux cubes et aux vues avec état.

Utilisez le nombre minimal de groupes nécessaires. Lorsque vous procédez à un regroupement à l'aide de la clé d'événement, un trop grand nombre de groupes est créé. Il est par conséquent conseillé d'effectuer des regroupements par région, minute, etc. N'effectuez pas de regroupement par client ni horodatage à moins d'avoir diminué au préalable la précision.

N'effectuez des calculs complexes qu'une seule fois. Au lieu de tronquer un horodatage plusieurs fois dans une vue, créez une colonne dans la vue dans laquelle le calcul est réalisé une seule fois, puis utilisez cette valeur dans les vues enfant.

© Copyright IBM Corp. 2007, 2014

TO_DATE() et les autres fonctions d'horodatage sollicitent énormément le processeur. Limitez le plus possible leur utilisation.

Concevez les fenêtres de manière appropriée. Tronquez les horodatages afin d'obtenir une granularité d'une minute ou d'une heure, en fonction du mode d'analyse des données que vous allez choisir. De plus, faites appel aux définitions de fenêtres adéquates. Par exemple, la forme longue `SUM(columnName) OVER (ORDER BY TruncatedTimeStampColumn RANGE INTERVAL '24' HOUR PRECEDING SLIDE INTERVAL '1' HOUR REFERENCE OPERATOR)` est préférable à la syntaxe simplifiée `MOV_SUM(NumericColumn)`.

Conservez les données d'une vue sans état. Ne tentez pas de conserver toutes les lignes en mémoire. Activez l'option **Persistance de la vue** pour stocker la vue et pouvoir la consulter ultérieurement à l'aide de l'option **Changement de niveau pour l'accès au détail**.

Avant de commencer

Vous devez disposer d'une autorisation de création sur les vues et d'une autorisation de lecture seule sur la table de flux de données qui alimente la nouvelle table.

Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Vue**.
4. Dans la fenêtre Nouvelle vue, sélectionnez le flux de données ou la vue dérivée pouvant servir de base à votre nouvelle vue. Utilisez le bouton **Parcourir** pour sélectionner des vues ou des flux de données dans un répertoire différent.
5. Cliquez sur **OK**.
6. Dans la fenêtre Configurer la vue, indiquez les attributs de votre choix.

Copie d'une vue

Vous pouvez également copier la définition d'une vue existante dans une nouvelle vue.

Procédure

1. Modifiez la vue à copier.
2. Changez le nom de la vue ainsi que les autres attributs qui diffèrent de la vue d'origine.
3. Cliquez sur **Enregistrer sous Nouvelle vue**.

Attributs de vue

Vous pouvez configurer une vue de façon à préciser les attributs dont vous avez besoin.

Tableau 100. Attributs d'une vue

Attribut	Description
Nom	Identifie l'objet de la vue. Ce nom ne peut comporter que des lettres et des chiffres. Il doit être unique parmi les vues, les flux de données, les tables de correspondance et les flux de données consolidés. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si la vue est activée (réception de nouvelles données) ou désactivée. Lorsqu'une vue est désactivée, tous les objets qui en dépendent sont également désactivés, notamment les règles, les alertes et les tableaux récapitulatifs.
Enregistrer dans	Indique le dossier dans lequel la table de flux de données sera stockée. Cliquez sur **Sélectionner un dossier** pour choisir ou créer un dossier. Si vous ne définissez pas de dossier, la valeur par défaut est **Dossiers publics**.
Description de la vue	Description facultative qui peut contenir tous les caractères de texte.
Ensemble de tâches	Tables de flux de données et vues à partir desquelles la vue est dérivée.
Faire glisser les zones en dehors de l'ensemble de tâches	Les zones sélectionnées à partir de l'ensemble de tâches correspondent aux colonnes à inclure dans la vue, notamment les colonnes dont les valeurs sont dérivées de formules. Ces colonnes génèrent la liste de sélection dans l'instruction SELECT sous-jacente. L'option **Grouper par** permet d'identifier les colonnes sur lesquelles regrouper les résultats. Il s'agit de la clause GROUP BY de l'instruction SELECT sous-jacente.
Garder dans la vue	Permet d'effectuer le suivi des informations sur les événements passés correspondant aux vues sans état. Pour en savoir davantage, voir «Conservation des événements dans des vues sans état», à la page 371.
Clause From	Clause permettant de joindre les informations de plusieurs tables et vues dans l'ensemble de tâches. Ces informations permettent de générer la clause FROM dans l'instruction SELECT sous-jacente.
Clause Where	Clause identifiant les informations source à inclure dans la nouvelle vue. Les flux de données qui ne respectent pas les spécifications ne sont pas inclus dans la vue. Ces informations permettent de générer la clause WHERE dans l'instruction SELECT sous-jacente. Même si les informations du flux de données sont supprimées, les vues dérivées sont mises à jour, mais elles n'incluent pas les informations du flux de données. Pour en savoir davantage, reportez-vous aux sections «Clause WHERE», à la page 343 et «Mise à jour des vues par le biais de la propagation d'événements», à la page 370.

Tableau 100. Attributs d'une vue (suite)

Attribut	Description
Clause Window	Définit les fenêtres pour l'agrégation des ensembles de lignes dans la vue. Pour en savoir davantage, voir Chapitre 38, «Fenêtres de requêtes», à la page 391.
Clause Order by	Trie les vues finales en fonction des noms de colonne ou des résultats d'expression. Ces informations permettent de générer la clause ORDER BY dans l'instruction SELECT sous-jacente.
Avancé	L'onglet **Avancé** permet de spécifier les options suivantes : • **Persistance de la vue** Permet de sauvegarder les données dans une base de données pour une analyse ultérieure. Pour en savoir davantage, voir «Conservation des vues dans une base de données», à la page 372. • **Changement de niveau pour l'accès au détail** Permet aux utilisateurs de visualiser les détails des données présentées par un objet de tableau de bord. Pour en savoir davantage, voir «Activation du changement de niveau pour l'accès au détail», à la page 374.

Contraintes sur les vues

Les vues IBM Cognos Real-time Monitoring possèdent des contraintes.

Ces contraintes sont les suivantes :

- Une vue peut être dérivée d'une table d'événements ou d'une autre vue.
- Une vue ne peut pas joindre plusieurs tables de flux de données. (Un flux de données consolidé est une union de deux tables de flux de données identiques. Pour plus d'informations, voir «Flux de données consolidés», à la page 369).
- Une vue peut joindre deux vues d'une même table de flux de données. (Une jointure synchronisée est une vue dérivée de plusieurs vues basées sur la même table de flux de données. Pour plus d'informations, voir «Jointures synchronisées»).
- Une vue peut joindre une table de flux de données et une ou plusieurs tables de correspondance.
- Une vue peut joindre une vue et une ou plusieurs tables de correspondance.
- Une vue ne peut pas être dérivée de tables de correspondance uniquement.

Jointures synchronisées

Une jointure synchronisée est une vue dérivée de deux vues basées sur la même table de flux de données d'origine.

Dans une jointure synchronisée, la vue finale contient des lignes qui sont la combinaison des mêmes tables de flux de données des vues source.

Figure 39. Vue d'une jointure synchronisée

Les jointures synchronisées sont toujours basées sur l'ID interne des table de flux de données ; la colonne de table de flux de données interne contient une condition de jointure implicite. Cependant, vous devez toujours définir une clause WHERE ou une condition de jointure dans la clause FROM pour éviter une éventuelle jointure croisée. Pour en savoir davantage, voir «Jointures croisées», à la page 341.

Une jointure synchronisée transmet en continu les résultats dans une vue sans état.

Restriction

Une jointure synchronisée requiert le traitement des événements dans l'ordre de leur arrivée.

Ainsi, l'option **Traitez les données selon l'ordre d'arrivée** doit être activée pour l'objet de flux de données.

Exemple de jointure synchronisée

L'illustration suivante présente un exemple de jointure synchronisée qui détermine le pourcentage des ventes totales par région.

La vue (*SalesTotal*) détermine les ventes totales de tous les événements alors que l'autre vue (*SalesByRegion*) détermine les totaux pour chaque région. La jointure synchronisée (*SalesRatiosByRegion*) détermine le pourcentage de chaque région en joignant les deux vues des ventes totales et en divisant le total global par les totaux des régions.

Figure 40. Exemple de jointure synchronisée

Notez que la jointure synchronisée ci-dessus est une jointure croisée, ce qui, dans cette situation, est acceptable. L'exemple suivant détermine le pourcentage des ventes de chaque partenaire commercial par rapport aux ventes totales :

Figure 41. Autre exemple de jointure synchronisée

Dans l'illustration précédente, vous ne souhaitez pas de jointure croisée, car elle crée une ligne pour chaque partenaire commercial de chaque région, y compris les régions dans lesquelles les commerciaux ne travaillent pas. La vue SalesRatiosPerRepRegion déclare, quant à elle, une jointure interne pour limiter les résultats par région.

Flux de données consolidés

Les flux de données consolidés sont des vues qui acceptent des événements à partir de deux sources d'événements différentes.

Pour plus d'informations, voir le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.

Vues agrégées

L'un des avantages clé des vues métier est la possibilité d'agréger des informations sur les tables de flux de données et de correspondance afin de les extraire, les analyser et les combiner en mesures métier pertinentes. Les vues agrégées contiennent au moins une définition de zone qui inclut une agrégation ou une clause GROUP BY.

Examinez, par exemple, cette vue simple qui effectue le suivi du nombre total des événements qui sont arrivés au cours de la dernière heure. A chaque arrivée d'un nouvel événement, la fonction MOV_COUNT() recalcule le nombre de tous les événements contenus dans la vue, fournissant un indicateur en temps réel de l'événement :

```
SELECT MOV_COUNT( *, HOUR, 1) AS "Data Streams in the
last hour"
    FROM Events
```

Vous pouvez accentuer la complexité de l'exemple précédent en limitant les événements affichés dans la vue. Par exemple, vous pouvez dénombrer uniquement les événements dont la valeur Status est Open :

```
SELECT MOV_COUNT( *, HOUR, 1) AS "Opens in the last hour"
    FROM Events
    WHERE Status='Open'
```

En combinant les informations agrégées avec un autre contexte, vous pouvez générer davantage d'indicateurs pertinents. Par exemple, cette vue présente le nombre d'événements dont la valeur Status est Open, puis les regroupe par Feature (fonction) :

```
SELECT Context.Topic AS Topic,
       MOV_COUNT( *, HOUR, 3) AS "Opens in 3 hours"
    FROM Events INNER JOIN Context ON Event.Feature =
Context.Feature
    WHERE (Status='Open')
    GROUP BY Feature
Feature        Opens in 3 hours
-------------- ----------------
Configuration  12
Install         3
Servers         6
```

Pour plus d'informations sur les fonctions d'agrégation et les fonctions d'ensembles mobiles (Moving set), voir «Types de fonction», à la page 125.

Mise à jour des vues par le biais de la propagation d'événements

Lorsqu'une vue reçoit un nouvel événement, elle tente de se mettre à jour avec les nouvelles informations. Si la mise à jour réussie, la vue avertit alors toutes les vues dépendantes de se mettre à jour également.

Cependant ces exceptions peuvent empêcher la vue de se mettre à jour :

- Si une vue est vide lors de la réception d'un événement et si elle reste vide après le traitement de l'événement, elle n'avertit jamais les vues dépendantes.
- Si une vue avec état se vide à la suite d'une suppression, par exemple si un événement existant est supprimé d'un ensemble de fonctions d'ensembles mobiles (Moving set), toutes les vues dépendantes sont averties qu'elles doivent également se mettre à jour.
- Une mise à jour peut provoquer le vidage d'une vue sans état, et celui de toutes les vues dépendantes. Si un événement ultérieur entraîne également le vidage d'une vue, celle-ci semblera ne pas avoir été mise à jour même si elle l'a été.
- Si un nouvel événement est supprimé, car il ne remplit pas certains critères, la vue avec état n'est pas mise à jour. Cependant, il publie encore une copie instantanée de lui-même pour toutes les vues dépendantes, ce qui peut pousser les fonctions d'ensembles mobiles (Moving set) dépendantes à mettre à jour leurs vues.

Si une erreur se produit lors du traitement d'un événement pendant la mise à jour d'une vue, toutes les lignes associées à l'événement entier sont supprimées, et la vue reste valide et activée.

Vues sans état et vues avec état

Dans IBM Cognos Real-time Monitoring, toutes les vues sont sans état ou avec état.

Les vues avec état contiennent les résultats des agrégations dérivées des événements passés d'une seule ligne. Une vue est avec état si elle :

- contient une fonction d'ensembles (Set) ou une fonction d'ensembles mobiles (Moving set) dans la clause SELECT, ou ;
- contient une clause GROUP BY (le cas échéant, chaque groupe contient une seule ligne), ou ;
- est dérivée d'une vue avec état.

Les vues sans état sont toutes les vues qui ne sont pas avec état.

En général, une vue sans état affiche les informations sur un seul événement, par exemple un bon de commande, alors qu'une vue avec état affiche les informations agrégées sur plusieurs événements, par exemple, le prix moyen de plusieurs bons de commande.

Initialisation des vues

Chaque vue conserve deux copies instantanées des données qu'elle contient.

Ces deux instantanés sont les suivants :

- Vue en cours

  Données actuellement dans la vue. S'il s'agit d'une vue avec état, la copie instantanée affiche toutes les lignes qu'elle comprend. S'il s'agit d'une vue sans état, elle affiche toutes les lignes correspondant au dernier événement, qui risque d'être un ensemble vide après l'agrégation.
- Vue récente

  Copie instantanée de la dernière vue en cours non vide. Si la vue possède une fenêtre d'ensembles mobiles (Moving set), la vue récente contient les dernières lignes non vides de la fenêtre. Par exemple, s'il s'agit d'une fenêtre de deux jours, la vue récente contient les dernières lignes 2 jours non vides, même si la vue en cours est vide. Une vue récente correspond à ce qui apparaît dans IBM Cognos Real-time Monitoring Workbench lors de la modification d'un objet qui affiche les résultats de la vue et à ce qui est utilisé par les vues dérivées lors de leur initialisation.

Lorsque vous créez ou activez une vue, elle est initialisée sur un état en fonction des données de la vue de base comme suit. Lorsqu'une vue est dérivée d'une :

- Vue avec état

  La nouvelle vue est initialisée avec les données de la copie instantanée de la vue en cours de la vue de base. Ci-après figure l'exemple d'une vue de base avec état qui effectue le suivi des ventes par région :

  ```
  SELECT
  region, SUM(sales) AS region_sales GROUP BY region
  ```

  Si vous dérivez une nouvelle vue de la vue sales_by_region, cette nouvelle vue est immédiatement renseignée à l'aide des données de la vue actuelle de la vue sales_by_region.

  ```
  SELECT SUM(region_sales) AS total_sales FROM sales_by_region
  ```
- Vue sans état

  La nouvelle vue est initialisée avec les données de la copie instantanée de la vue récente de la vue de base. Ci-après figure l'exemple d'une vue sans état. La copie instantanée actuelle de cette vue est vide si aucune vente n'est supérieure à 1 000 000.

  ```
  SELECT region,
  sales WHERE
  sales > 1000000
  ```

  Cependant, une vue dérivée de cette vue est initialisée avec les données de la copie instantanée de la vue récente de cette vue.
- Table de flux de données

  La nouvelle vue est vide. Les tables de flux de données ne conservent pas les copies instantanées.

Conservation des événements dans des vues sans état

Par défaut, une vue sans état contient uniquement les lignes qui correspondent au dernier événement ayant rempli les conditions de la vue. Les lignes des événements précédents sont supprimées.

Si l'événement ne remplit pas les conditions, cette vue est vide. L'option **Garder dans la vue** vous permet de spécifier un ensemble d'informations sur des événements non vides récents à conserver dans la vue.

Utilisez cette option pour insérer les événements récents dans les situations suivantes :

- Lors de l'affichage du contenu de la vue dans l'onglet **Résultats**. Cet onglet affiche les lignes en cours dans la vue.
- Lors de l'utilisation d'applications externes qui reçoivent la vue en tant que données en temps réel. Lorsqu'elle est activée, cette option permet à l'application externe d'effectuer des analyses de tendances ou historiques.

Procédure

1. Modifiez la vue que vous souhaitez conserver.
2. Dans la fenêtre Configurer la vue, cochez la case **Garder dans la vue**.
3. Entrez le compte ou la plage de temps des événements à conserver.
   - Un compte d'événements correspond au nombre maximal d'événements non vides à conserver. La vue supprime les lignes des événements les moins récents qui ne correspondent pas à la taille indiquée.
   - Un intervalle de temps définit un ensemble des événements les plus récents. Le nombre des événements de la vue varie en fonction du nombre d'événements présents dans l'intervalle lors de la mise à jour de la vue. Par exemple, si un événement arrive, mais ne correspond pas aux critères de la vue, il en est exclu. Toutefois, la vue recalcule l'intervalle correspondant à cette heure.

   L'ensemble d'événements a été déterminé lors de l'insertion du dernier événement et non pas à l'heure actuelle. Par exemple, un intervalle d'une heure indique tous les événements arrivés dans la vue durant l'heure précédant la dernière mise à jour. Si aucun événement n'a été inséré au cours du dernier jour, il se peut que la vue continue d'afficher les événements de la journée précédente survenus sur une période d'une heure. Cependant, dès qu'un nouvel événement arrive dans la vue, tous ces événements sont supprimés.

   Pour plus d'informations sur l'agrégation d'ensembles d'événements récents, voir «Fonctions d'ensembles mobiles», à la page 127.

Conservation des vues dans une base de données

IBM Cognos Real-time Monitoring peut conserver les données de la vue métier dans un SGBD externe en vue de la production ultérieure d'un rapport par des outils tiers.

Les informations de la table permettent aux outils de génération de rapports de recréer une copie instantanée complète de la vue. Au moment de la sauvegarde, les informations de la vue s'inscrivent dans une table du SGBD à un taux conforme à une politique définie.

Cognos Real-time Monitoring peut créer la table automatiquement ou vous pouvez prédéfinir la table dans le SGBD. Pour en savoir davantage, voir «Attributs de sauvegarde des vues», à la page 373.

Avant de commencer

Un spécialiste de l'application doit d'abord définir un agent JDBC dans le SGBDR qui stocke les données de la vue. Pour en savoir davantage, voir «Agents JDBC

(Java Database Connectivity)», à la page 238.

Procédure

1. Modifiez la vue que vous souhaitez conserver.
2. Dans la fenêtre Configurer la vue, cliquez sur l'onglet **Avancé**.
3. Sous **Persistance de la vue**, cliquez sur **Définir**.
4. Complétez les zones dans la fenêtre Définir la persistance de la vue. Pour en savoir davantage, voir «Attributs de sauvegarde des vues».

   La vue commence à conserver les données dès que cette option est activée.

Attributs de sauvegarde des vues

Vous pouvez spécifier ces attributs lorsque vous sauvegardez les données de la vue métier dans un SGBD externe.

Tableau 101. Attributs de sauvegarde des vues

Attribut	Description
Agent JDBC	Agent du SGBDR défini dans la tâche.
Table	Table cible du SGBDR destinée à recevoir les données sauvegardées. Si vous omettez ce nom, la table a le même nom que la vue métier. Les colonnes de la table cible doivent avoir les mêmes noms que les colonnes de la vue, s'afficher dans le même ordre que dans la vue et être au moins de la même largeur que les colonnes de la vue. Si la largeur d'une colonne de la table cible est inférieure, le SGBDR tronque les données pour s'ajuster ou génère une erreur. De même, la table cible doit prendre en charge des lignes au moins aussi longues que celles de la vue métier.
Créer cette table si elle n'existe pas	Indique que le spécialiste de l'application ou l'administrateur de base de données n'a pas encore créé la table nommée. Si cette option est sélectionnée, IBM Cognos Real-time Monitoring tente de créer la table cible à l'aide de la commande CREATE TABLE dans l'espace de table par défaut de l'utilisateur JDBC.
Politique de persistance	La sauvegarde varie selon que la vue est sans état ou avec état. Les vues avec état sont écrites en tant que copies instantanées qui sont sauvegardées en fonction de l'intervalle spécifié, et seules les copies instantanées sont sauvegardées. Les vues sans état sont également écrites en tant que copies instantanées, mais elles sont aussi enregistrées pour que les conditions existantes entre les copies soient également capturées.
Nombre d'événements	Spécifie le nombre d'événements à écrire dans la base de données. Par exemple, si le nombre d'événements est défini sur 10, alors 10 événements doivent se produire avant d'être écrits dans la base de données. Définir ce nombre sur 1 provoque une écriture pour chaque événement. Un événement constitue une ligne dans une base de données. Les événements sont représentés par des objets de flux de données dans Cognos Real-time Monitoring.
Intervalle	Spécifie la fréquence des écritures dans la base de données.
Désactiver lorsque le nombre d'erreurs consécutives suivant est atteint	Indique le nombre d'erreurs consécutives à écrire dans le journal des erreurs avant la désactivation de la sauvegarde. Cette option empêche le serveur de répéter la même erreur.

Colonnes des vues à sauvegarder

La base de données reçoit toutes les colonnes et les lignes actuellement contenues dans la vue.

En outre, chaque ligne contient ces colonnes internes supplémentaires :

Tableau 102. Colonnes internes

Colonne	Description
VC_EVENT_ID	L'identificateur d'événement identifie l'événement ayant produit la toute dernière ligne insérée dans la vue.
VC_LATEST_EVENT_ID	L'identificateur du dernier événement identifie le dernier événement ayant provoqué la mise à jour de la vue, même si les données de cet événement ne sont peut-être pas incluses dans la vue.
VC_TIMESTAMP	L'horodatage d'événement identifie le moment auquel le dernier événement a été inséré dans la vue.

Activation du changement de niveau pour l'accès au détail

Vous pouvez créer des vues offrant la possibilité d'accéder aux détails d'un graphique.

Les utilisateurs peuvent se servir de cette option pour visualiser les détails des données présentées par un objet de tableau de bord. Les spécifications pour le **Changement de niveau pour l'accès au détail** sont les suivantes :

- La source de données de l'objet de tableau de bord doit être un cube.
- La vue sur laquelle le cube est conçu doit être sans état. Pour en savoir davantage, voir «Conservation des événements dans des vues sans état», à la page 371.
- Les dimensions du cube doivent provenir d'une seule base de données externe et cette base de données doit être identique aux mesures du cube.
- Si une vue comporte plusieurs tables, vous pouvez utiliser une table sauvegardée comme table de changement de niveau pour l'accès au détail. Pour en savoir davantage, voir «Conservation des vues dans une base de données», à la page 372.

Les limitations de l'option **Changement de niveau pour l'accès au détail** sont les suivantes :

- Les espaces et caractères spéciaux ne sont pas pris en charge dans les noms des tables et colonnes des vues utilisées pour un changement de niveau avec persistance de la vue.
- Il est conseillé de mettre le nom des colonnes des vues en caractères majuscules pour l'utilisation du changement de niveau. Sinon, l'accès au détail risque de ne pas fonctionner.
- Un changement de niveau avec filtre sur une colonne de dimension de type TIMESTAMP ne fonctionne pas si le contexte de la dimension est stocké dans une base de données Oracle.
- Si le créateur du cube a filtré l'accès à la vue de fait, le cube n'est pas activé pour le changement de niveau.
- Le changement de niveau pour l'accès au détail n'est pris en charge que sur les cubes construits sur une vue sans état.

Avant de commencer

Un spécialiste de l'application doit d'abord définir un agent JDBC dans le SGBDR qui stockera les données de la vue. Pour plus d'informations, voir «Agents JDBC (Java Database Connectivity)», à la page 238.

Procédure

1. Ouvrez l'éditeur de vue dans une vue pour laquelle vous souhaitez activer l'option **Changement de niveau pour l'accès au détail**.
   - Dans une vue existante, sélectionnez **Modifier cette vue**.
   - Lors de la création d'une vue, sélectionnez **Nouvelle vue**.
2. Cliquez sur l'onglet **Avancé**, puis sous **Changement de niveau pour l'accès au détail**, cliquez sur **Définir**.
3. Dans la zone **Agent JDBC**, sélectionnez l'agent du SGBDR.
4. Dans la zone **Table**, spécifiez la table du SGBDR à utiliser pour obtenir des informations sur le changement de niveau pour l'accès au détail. Il peut s'agir d'une table de faits ou de la table que vous avez spécifiée dans la zone **Persistance de la vue**.
5. Cliquez sur **OK**.
6. Enregistrez la vue.

Données IBM Cognos Real-time Monitoring dans IBM Cognos Business Intelligence

Vous pouvez faire appel à un cube ou une vue IBM Cognos Real-time Monitoring comme source de données dans IBM Cognos Business Intelligence.

Vous pouvez utiliser les données de Cognos Real-time Monitoring dans Business Intelligence. Par exemple, vous pouvez visualiser les données Real-time Monitoring dans Query Studio ou créer et publier un package de rapports dans Framework Manager, puis afficher ces rapports dans Cognos Connection.

Affichage des données IBM Cognos Real-time Monitoring dans IBM Cognos Business Intelligence

Vous avez la possibilité d'afficher les données IBM Cognos Real-time Monitoring dans IBM Cognos Business Intelligence.

Les types de données disponibles sont les suivants :

- Cubes Cognos Real-time Monitoring.
- Vues Cognos Real-time Monitoring.

Avant de commencer

Avant de réaliser cette procédure, effectuez les actions suivantes :

- Vérifiez que le serveur Cognos Real-time Monitoring fonctionne.

  Conseil : Si vous parvenez à ouvrir Cognos Real-time Monitoring dans un navigateur Web, cela signifie que le serveur est en cours d'exécution.
- Vérifiez que le serveur Cognos Business Intelligence fonctionne.

Conseil : Utilisez Cognos Configuration pour vérifier que le serveur Cognos Business Intelligence est en cours d'exécution. Pour plus d'informations, consultez l'aide en ligne de Cognos Configuration.

- Configurez le code d'accès unique entre Cognos Real-time Monitoring et Cognos Business Intelligence. Pour plus d'informations, consultez l'aide en ligne d'IBM Cognos Real-time Monitoring Workbench.

Procédure

1. Dans IBM Cognos Administration, créez une connexion de source de données vers un cube Real-time Monitoring. Pour en savoir plus à ce sujet, consultez l'aide en ligne de la console d'administration IBM Cognos.
2. Créez un nouveau projet dans Framework Manager. Pour plus d'informations sur la création d'un projet dans Framework Manager, consultez l'aide en ligne de Framework Manager.
3. Dans l'assistant de métadonnées de Framework Manager, vérifiez d'avoir sélectionné la source de données pour laquelle vous avez établi une connexion à l'étape 1.
4. Dans l'assistant de publication de Framework Manager, créez et publiez le pack vers le serveur Business Intelligence.
5. Vous pouvez à présent visualiser les données dans Business Intelligence. Par exemple, vous pouvez afficher les métadonnées dans Query Studio ou créer des rapports dans Framework Manager, puis les afficher dans Cognos Connection.

Chapitre 37. Services Web

Un service Web sert d'interface à une application s'exécutant sur un serveur d'applications Web. Ce service peut être un simple script de recherche de base de données ou bien un produit complexe d'intégration des applications de l'entreprise (EAI), tels que les produits fournis par Siebel ou SAP.

IBM Cognos Real-time Monitoring se connecte aux services Web pour effectuer les tâches suivantes :

- Recevoir des données au format XML. Pour en savoir davantage, voir «Flux de données de service Web».
- Récupérer des tables de correspondance. Pour en savoir davantage, voir «Tables de correspondance pour les services Web», à la page 381.
- Publier des messages d'alerte :
  - Au profil de diffusion d'un abonné. Pour en savoir davantage, voir «Onglet Profils de diffusion», à la page 356.
  - Un à un comme initié par un utilisateur affichant le message dans IBM Cognos Real-time Monitoring Dashboard. Pour en savoir davantage, voir «Processus externes des services Web», à la page 386.

Flux de données de service Web

Les services Web publient des données au format XML par le biais du protocole HTTP directement vers les serveurs IBM Cognos Real-time Monitoring. Tous les éditeurs de services Web utilisent la même URL. Ainsi, le nom du flux de données qui reçoit les données ainsi que le compte Cognos Real-time Monitoring qui a accès à la table de flux de données sont encodés en XML. Lorsque les serveurs reçoivent les données, ils analysent le code XML, décodent les données, puis les insèrent dans la table de flux de données identifiée.

Lorsque vous configurez un flux de données pour un service Web, vous pouvez choisir l'un des deux types de flux de données suivants : service Web ou interrogation du service Web. Pour en savoir davantage, voir «Attributs des flux de données pour les services Web», à la page 378 ou «Attributs des flux de données pour les services Web d'interrogation», à la page 379. Ces deux types sont différents. Lorsque vous sélectionnez **Service Web**, les données sont envoyées à Cognos Real-time Monitoring à partir du service Web. Lorsque vous sélectionnez **Interrogation du service Web**, les données sont extraites à partir du service Web spécifié à intervalles réguliers.

Figure 42. Flux de données de service Web

© Copyright IBM Corp. 2007, 2014

Attributs des flux de données pour les services Web

Chaque table de flux de données associée à un service Web comporte plusieurs attributs.

Ces attributs sont décrits dans le tableau ci-dessous.

Attribut	Description
Nom	Identifie la table. Il s'agit du nom utilisé par les vues métier qui dépendent de cette table. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Description	Description facultative qui peut contenir tous les caractères de texte.
Enregistrer dans	Indique le dossier dans lequel enregistrer l'agent. La valeur par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Sinon, si les événements peuvent être traités dans n'importe quel ordre, désactivez cette option. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive le flux de données lorsque le nombre d'erreurs consécutives défini est atteint. Par exemple, si vous avez défini cet attribut sur 5, l'événement est désactivé après 5 erreurs consécutives. En revanche, si 4 erreurs surviennent, puis aucune erreur, puis 2 nouvelles erreurs, le flux de données reste activé. La valeur par défaut est désactivée.
Informations sur la colonne	Les zones **Informations sur la colonne** définissent les colonnes de la table de flux de données et portent le même nom que les zones du message XML, tel que décrit dans le fichier WSDL correspondant au flux de données.
Intervalle de suppression de l'état	Cet onglet comporte plusieurs options de suppression des données d'événement. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.

Attributs des flux de données pour les services Web d'interrogation

Chaque table de flux de données associée à un service Web d'interrogation comporte plusieurs attributs.

Ces attributs sont décrits dans le tableau ci-dessous.

Avertissement : Lorsque vous définissez des calendriers, l'heure est basée sur l'heure du serveur et non sur l'heure locale.

Tableau 103. Attributs des flux de données pour les services Web d'interrogation

Attribut	Description
Nom	Identifie la table. Il s'agit du nom utilisé par les vues métier qui dépendent de cette table. Ce nom doit être unique parmi les vues, les flux de données, les tables de correspondance et les flux de données consolidés. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception et transmission de données) ou désactivé (pas de réception ni de transmission de données).
Description	Description facultative de la table.
Enregistrer dans	Indique le dossier dans lequel enregistrer l'agent. La valeur par défaut est **Dossiers publics**. Cliquez sur **Sélectionner un dossier** pour choisir un dossier.
Agent de service Web	Agent utilisé par le flux de données. Cliquez sur le bouton **Modifier** pour sélectionner l'agent de service Web.
Type de port	(Facultatif) Indique le type de port requis pour clarifier les opérateurs portant le même nom. La nécessité de fournir un type de port est basée sur le langage WSDL que vous utilisez.
Méthode	Méthode du service Web à utiliser. Lorsque le service propose plusieurs méthodes, vous devez sélectionner celle que vous utiliserez.
Activer la récupération	Lorsque la récupération est activée, les données provenant d'un événement arrivé après le dernier point de contrôle sont consignées. Cette option permet de rétablir l'état du système en cas d'arrêt anormal des serveurs. Pour plus d'informations, voir les rubriques relatives aux informations sur la reprise et les points de contrôle dans le *Guide d'utilisation d'IBM CognosReal-time Monitoring Workbench*.
Traitez les données selon l'ordre d'arrivée	Sélectionnez cette option si les événements doivent être traités en suivant l'ordre de leur réception. Sinon, si les événements peuvent être traités dans n'importe quel ordre, désactivez cette option. Pour lier les événements d'une vue, vous devez les traiter dans l'ordre. Activez cette option pour lier les événements.
Désactiver le flux de données lorsque ce nombre d'erreurs consécutives est atteint	Désactive l'événement lorsque le nombre d'erreurs consécutives défini est atteint. Par exemple, si cet attribut est configuré sur 5, l'événement est désactivé après 5 erreurs consécutives. En revanche, si 4 erreurs surviennent, puis aucune erreur, puis 2 nouvelles erreurs, le flux de données reste activé. La valeur par défaut est **Ne pas désactiver**.

Tableau 103. Attributs des flux de données pour les services Web d'interrogation (suite)

Attribut	Description
Traiter toutes les lignes de l'ensemble de résultats comme un seul événement	Toutes les lignes renvoyées dans l'ensemble de résultats sont considérées comme un événement unique. Sinon, chaque ligne renvoyée de la table est considérée comme un événement distinct.
Racine XPath	Identifie la racine XPath des éléments répétitifs dans la sortie qui encapsule une ligne de données.
Informations sur la colonne	Les zones **Informations sur la colonne** définissent les colonnes dans la table de flux de données. Elles portent le même nom que les zones du message XML, comme décrit dans le fichier WSDL du flux de données.
Arguments d'entrée	Les zones **Arguments d'entrée** correspondent aux valeur fournies dans l'entrée à la méthode, si cette dernière a besoin de valeurs d'entrée. Si la méthode ne dispose d'aucun argument d'entrée, il n'est pas nécessaire de définir des zones d'arguments d'entrée.
Intervalle de suppression de l'état	Cet onglet contient plusieurs options permettant de supprimer les données d'événements sauvegardées. Pour en savoir davantage, voir «Intervalle de suppression de l'état», à la page 61.
Interrogation	Cette table contient plusieurs options permettant de définir la fréquence d'interrogation du service Web à la recherche de nouvelles données. Les options sont les suivantes : • **Interroger selon un intervalle** Permet de définir l'intervalle d'interrogation. La valeur par défaut est 600 secondes. • **Interroger selon un calendrier** Interroge le service Web selon un calendrier. Lorsque vous sélectionnez cette option, la fonction de calendrier est activée.
Désactiver le flux de données lorsque ce nombre d'erreurs d'interrogation consécutives est atteint	Désactive le flux de données lorsque le nombre d'erreurs d'interrogation consécutives est atteint. Par exemple, si vous avez défini cet attribut sur 5, le flux de données est désactivé après 5 erreurs d'interrogation consécutives. En revanche, si 4 erreurs surviennent, puis aucune erreur, puis 2 nouvelles erreurs, le flux de données reste activé.

Création d'un flux de données de service Web

Avant de créer une table de flux de données de service Web, vous devez créer une autorisation sur les tables.

Pour en savoir davantage, reportez-vous aux sections Chapitre 23, «Autorisations», à la page 295 et «Autorisation de création d'objets», à la page 299.

Procédure

1. Créez un flux de données de service web dans IBM Cognos Real-time Monitoring Workbench.
2. (Facultatif) Créez un compte utilisateur IBM Cognos Real-time Monitoring pour le service web à utiliser lors de la publication de l'événement. Ce compte doit disposer au moins d'une autorisation de lecture et d'écriture sur la table de flux de données pour publier cette dernière.
3. Récupérez la définition WSDL de l'événement à l'aide du protocole HTTP.

```
http://<host:port>/cognos/realtime/wsdl/eventstream.wsdl
```

Le fichier eventstream.wsdl décrit tous les objets de flux de données définis pour les services Web dans Cognos Real-time Monitoring. Consultez la documentation de WSDL pour obtenir la description des éléments et des attributs XML.

4. Créez le client de service web respectant la définition WSDL à l'étape 3.

   Cet exemple de syntaxe XML transfère des données vers le flux de données OrderWSEvent et indique le compte utilisé pour accéder au flux de données (WSInputAccount).

   Il existe quatre colonnes de données définies dans l'élément <OrderWSEventData>.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <OrderWSEvent_input xmlns="http://cognos.obi.com">
         <cqesi>
            <eventname>OrderWSEvent</eventname>
            <username>WSInputAccount</username>
            <password>wspwd</password>
         </cqesi>
         <OrderWSEventData>
            <ProdName>Plywood</ProdName>
            <OrderQuantity>150</OrderQuantity>
            <OrderTotal>987.34</OrderTotal>
            <IsBackordered>true</IsBackordered>
         </OrderWSEventData>
      </OrderWSEvent_input>
   </soapenv:Body>
</soapenv:Envelope>
```

5. Publiez les flux de données vers l'URL identifiée dans le fichier eventstream.wsdl, semblable à l'exemple suivant :

```
http://<host:port>/cognos/realtime/webservice/eventstream.wsdl
```

   Si plus tard, vous changez l'ordinateur qui héberge les serveurs Cognos Real-time Monitoring, veillez à interroger de nouveau le fichier WSDL afin d'identifier l'URL correcte.

Tables de correspondance pour les services Web

Les vues métier demandent des lignes à partir d'une table de correspondance liées à une ou plusieurs valeurs d'entrée, telles qu'une liste de fournisseurs d'un élément dans lequel l'ID de l'article représente l'entrée. Cette entrée est ensuite transmise à l'application de service Web via l'agent en tant que XML. L'application renvoie ensuite une ou plusieurs lignes de données XML, qui sont mappées dans la table de correspondance. La table transmet les données requises à la vue métier appropriée.

Figure 43. Tables de correspondance pour les services Web

Les agents de service Web IBM Cognos Real-time Monitoring sont synchrones. Ils récupèrent les données de la table de correspondance suite à une demande spécifique. Lors de la demande de données, l'agent fait appel au protocole de liaison SOAP pour communiquer avec l'application via une connexion HTTP. L'application renvoie ensuite une ou plusieurs lignes de données XML, suivant le format du langage WSDL, le format de type de document. Le style RPC WSDL n'est pas pris en charge.

Pour plus d'informations sur les agents de table de correspondance, voir «Agents de services Web», à la page 41.

Pour créer une table de correspondance à l'aide des données d'un service Web, vous devez disposer des éléments suivants :

- Autorisation de création sur les tables. Pour en savoir davantage, reportez-vous aux sections Chapitre 23, «Autorisations», à la page 295 et «Autorisation de création d'objets», à la page 299.
- Agent de service Web qui alimente la table. Pour en savoir davantage, voir «Agents de services Web», à la page 41.
- Droits d'accès en lecture seule sur l'agent.

Les attributs d'une table de correspondance pour les services Web sont les suivants :

Tableau 104. Attributs d'une table de correspondance pour les services Web

Attribut	Description
Nom	Nom de la table de correspondance. Ce nom doit être unique parmi l'ensemble des objets (notamment les tableaux de bord et les objets de tableau de bord) d'un même dossier. Le dossier racine contient également des objets globaux, tels que des utilisateurs, des rôles et des activités métier. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Etat	Indique si l'objet est activé (réception de données) ou désactivé (pas de réception de données).
Description	Description facultative qui peut contenir tous les caractères de texte.

Tableau 104. Attributs d'une table de correspondance pour les services Web (suite)

Attribut	Description
Enregistrer dans	Indique le dossier dans lequel enregistrer l'agent. La valeur par défaut est **Dossiers publics**. Cliquez sur le bouton **Sélectionner un dossier** pour désigner le dossier souhaité.
Agent de service Web	Agent de service Web existant qui se connecte à une application de service Web. Créez un agent depuis l'onglet d'administration du plan de travail. Pour en savoir davantage, voir «Agents de services Web», à la page 41. Cette valeur n'est pas modifiable.
Méthode	Méthode du service Web à utiliser. Lorsque le service propose plusieurs méthodes, vous devez sélectionner celle que vous utiliserez. Cette valeur n'est pas modifiable dans cette édition.
Désactiver la table de correspondance lorsque ce nombre d'erreurs consécutives	Nombre d'erreurs consécutives pouvant se produire avant que le système ne désactive cette table. Une fois désactivée, une table de correspondance doit être réactivée manuellement. La valeur par défaut est 5.
Racine XPath	Identifie la racine XPath des éléments répétitifs dans la sortie, en général /Envelope/Body. Ce chemin est ajouté devant tous les chemins de la liste **Nom de la zone de sortie**.
Résultats	Colonnes recevant les informations du service Web. Pour plus d'informations sur les colonnes de sortie, voir «Colonnes de sortie», à la page 384.
Entrées	Colonnes contenant les données qui identifient les éléments à rechercher dans la requête. Pour plus d'informations sur les colonnes d'entrée, voir «Colonnes d'entrée», à la page 384.
Mise en cache des données	Pour plus d'informations sur la mise en cache, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

Création d'une table de correspondance de services Web

Utilisez la procédure suivante pour créer une table de correspondance des données d'un service web.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur l'onglet **Plan de travail**.
2. Cliquez sur **Activités**.
3. Cliquez sur **Créer** > **Table de correspondance**.
4. Dans la fenêtre Nouvelle table de correspondance, sélectionnez **Service Web** comme type de source.
5. Choisissez un agent **Service Web** existant.
6. Sélectionnez la méthode du service à utiliser. Chaque service web propose une ou plusieurs méthodes permettant d'accéder aux données fournies conformément à sa définition de fichier WSDL (emplacement de l'URL que vous avez spécifiée lors de la création de l'agent de service web). Choisissez la méthode qui exécutera la requête dont a besoin votre contexte.
7. Définissez les colonnes qui recevront les informations du service web, c'est-à-dire, les colonnes de sortie. Par défaut, l'éditeur définit une colonne pour chaque élément renvoyé par la méthode. Pour plus d'informations sur la définition de ces colonnes, voir «Colonnes de sortie», à la page 384.
8. Définissez les colonnes qui contiennent les données identifiant les éléments à rechercher dans la requête, c'est-à-dire, les colonnes d'entrée. Par défaut,

l'éditeur définit une colonne pour chaque élément renvoyé par la méthode. Pour plus d'informations sur les colonnes d'entrée, voir «Colonnes d'entrée».

9. Indiquez le nombre de résultats à mettre en cache éventuellement. Pour plus d'informations sur la mise en cache, voir «Mise en cache des requêtes de tables de correspondance», à la page 269.

   Enregistrez la table des services web avec l'état activé.

Colonnes de sortie

Les colonnes Sortie reçoivent les informations du service Web et définissent la table qui recevra les données. L'éditeur définit automatiquement une colonne pour chaque élément renvoyé par la méthode.

Chaque colonne possède les attributs suivants :

Tableau 105. Attributs des colonnes de sortie

Attribut	Description
Colonne	Nom de la colonne de la table qui contient les résultats renvoyés par le service Web. Par défaut, ce nom est identique à celui de l'élément contenu dans Nom de la zone de sortie. Vous pouvez attribuer n'importe quel nom valide.
Nom de la zone de sortie	Identifie l'élément XML renvoyé par le service. Vous pouvez afficher le chemin d'accès complet à l'élément à l'aide du bouton Masquer/Afficher. Cet attribut n'est pas modifiable.
Type de données XSD	Identifie le type de données de l'élément XML. Seuls les types de données de base sont pris en charge : nombres, chaînes, dates et valeurs booléennes. Les types complexes comme ANY et ARRAY et les types MIME ne sont pas pris en charge. Cet attribut n'est pas modifiable.
Type de données	Type de données IBM Cognos Real-time Monitoring de la colonne de la table. Sélectionnez un type de données approprié aux données renvoyées.
Mise en forme	Formate les valeurs Chaîne, Date-heure ou Décimal renvoyées. Cette option n'est pas disponible pour les autres types de données. Pour plus d'informations sur les types de données, voir Chapitre 9, «Types de données», à la page 69.

Pour exclure des colonnes du résultat, ouvrez la boîte de dialogue **Masquer/Afficher**, puis désélectionnez les zones correspondantes.

Colonnes d'entrée

Les colonnes Entrée transmettent des informations au service Web pour qu'il identifie les données à renvoyer (les sorties). L'éditeur définit automatiquement une colonne pour chaque élément identifié par la méthode. Chaque colonne possède les attributs suivants :

Tableau 106. Attributs des colonnes d'entrée

Attribut	Description
Colonne	Nom de la colonne qui contient les informations transmises à la requête du service Web. Par exemple, il peut contenir un ID identifiant le produit à rechercher. Cette colonne est renseignée par la vue métier qui requiert les informations de la table de correspondance. Par défaut, son nom est identique à celui de l'élément contenu dans Nom de la zone d'entrée. Vous pouvez attribuer n'importe quel nom valide.
Nom de la zone d'entrée	Identifie l'élément XML transmis au service. Vous pouvez afficher le chemin d'accès complet à l'élément à l'aide du bouton Masquer/Afficher. Cet attribut n'est pas modifiable.
Modèle de substitution de la chaîne	Chaîne à transmettre au service qui contient les valeurs insérées par la vue métier demandant les informations. Pour en savoir davantage, voir «Modèles de substitution de la chaîne».
Type de données XSD	Identifie le type de données de l'élément XML. Seuls les types de données de base sont pris en charge : nombres, chaînes, dates et valeurs booléennes. Les types complexes comme ANY et ARRAY et les types MIME ne sont pas pris en charge. Cet attribut n'est pas modifiable.
Type de données	Type de données IBM Cognos Real-time Monitoring de l'attribut **Colonne**. Sélectionnez un type de données approprié aux données à transmettre.
Mise en forme	Formate les valeurs Chaîne, Date-heure ou Décimal renvoyées. Cette option n'est pas disponible pour les autres types de données. Pour en savoir davantage, voir Chapitre 9, «Types de données», à la page 69.

Pour exclure les colonnes superflues de la requête, ouvrez la boîte de dialogue **Masquer/Afficher**, puis désélectionnez les zones correspondantes.

Modèles de substitution de la chaîne

Utilisez un modèle de substitution de la chaîne lorsque le service web nécessite une chaîne avec les données de recherche incorporées.

L'option **Modèle de substitution de la chaîne** vous permet d'associer les paramètres du flux de données de sorte qu'ils soient transmis au service web sous la forme d'une zone unique.

La substitution de chaîne est facultative. Vous utilisez ces modèles uniquement en fonction de votre service web.

Procédure

1. Cliquez sur **...** pour ouvrir l'éditeur de modèle.
2. Saisissez l'expression du modèle suivie d'un signe égal (=) et d'un point d'interrogation (?) correspondant au nom de la colonne à remplacer. Pour associer plusieurs noms de colonnes, entrez une perluète (&) entre chaque expression.
3. Sous **Zones de substitution**, saisissez le nom que vous souhaitez utiliser pour chaque zone de substitution.
4. Enregistrez le modèle pour mettre à jour les zones **Colonne** de la section **Entrées**. Les noms que vous avez attribués aux zones de substitution sont alors utilisés.

Processus externes des services Web

Les processus externes des services Web sont des méthodes qui reçoivent les documents XML qui décrivent le message d'alerte ou l'objet de tableau de bord qu'un utilisateur IBM Cognos Real-time Monitoring Dashboard affiche et qu'il a envoyé au service. Le document XML décrit toutes les données de l'élément affiché par l'utilisateur.

Publication d'un élément vers un service Web externe

Vous pouvez publier un élément vers un service Web externe.

Procédure

1. Définissez le processus externe pour recevoir le message dans IBM Cognos Real-time Monitoring Workbench. Pour en savoir davantage, voir «Création d'un processus externe».
2. Envoyez l'élément à partir de Cognos Real-time Monitoring Dashboard. Lors de l'affichage d'un message d'alerte, d'une tâche ou d'un objet de tableau de bord, sélectionnez **Agir > Initier le processus**, puis le processus. Le service externe reçoit l'élément sous la forme d'un document XML et le traite. Pour en savoir davantage, voir «Implémentation du service externe», à la page 387.

Attributs de processus externe

Chaque processus externe est associé à des attributs.

Ces attributs sont décrits dans le tableau ci-dessous.

Tableau 107. Attributs de processus externe

Attribut	Description
Nom	Nom du processus externe qui s'affiche dans la boîte de dialogue Initier le processus d'IBM Cognos Real-time Monitoring Dashboard. Ce nom doit être unique parmi les processus externes. Pour en savoir davantage, voir «Espace-noms d'objets», à la page 281.
Description	Description facultative qui peut contenir tous les caractères de texte. Cette description s'affiche dans la boîte de dialogue Initier le processus de Real-time Monitoring Dashboard.
Etat	Indique si le processus est activé (envoi de documents XML) ou désactivé (pas d'envoi de documents).
Adresse URL du service Web	Emplacement HTTP de l'application fournissant le service RPC (liaison SOAP). Notez que les messages de style DOC ne sont pas pris en charge.
Méthode	Méthode du service Web à utiliser.
Nom d'utilisateur	Compte à utiliser pour se connecter au service (facultatif).
Mot de passe	Mot de passe du compte utilisateur (facultatif).

Création d'un processus externe

Vous pouvez définir un processus externe à partir de l'onglet **Console d'administration** d'IBM Cognos Real-time Monitoring Workbench.

Avant de commencer

Avant de créer un processus externe des services web, vous devez disposer des éléments ci-dessous :

- Une méthode de service Web externe définie pour recevoir la méthode publiée. Vous devez connaître l'URL pour vous connecter au service, le nom de la méthode qui reçoit le message, et tous les noms d'utilisateur ou de compte ainsi que les mots de passe requis par le service. Pour en savoir davantage, voir «Implémentation du service externe».
- Une autorisation de **création** sur les processus externes. Pour en savoir davantage, voir «Autorisation de création d'objets», à la page 299.

Procédure

1. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur **Console d'administration**.
2. Cliquez sur **Processus externes**.
3. Cliquez sur **Nouveau processus**.
4. Définissez les attributs du processus.
5. Enregistrez le processus

   Pour utiliser le processus, dans IBM Cognos Real-time Monitoring Dashboard, sélectionnez **Agir > Initier le processus**, puis le processus.

   Pour afficher et utiliser un processus dans Cognos Real-time Monitoring Dashboard, les utilisateurs doivent disposer d'une autorisation de lecture sur ce processus. Vous pouvez attribuer des autorisations sur le nouvel objet en cliquant sur Droits dans la liste des processus externes ou bien un administrateur peut accorder aux utilisateurs un accès en lecture sur la classe des processus externes. Pour en savoir davantage, voir «Droits d'accès», à la page 295.

Implémentation du service externe

Vous pouvez implémenter le service web externe.

Pour ce faire, suivez les étapes ci-dessous.

Procédure

1. Définissez le service web pour recevoir un message de liaison SOAP avec les zones de l'action externe XSD.
2. Créez un fichier WSDL (fichier de définition) en suivant les conditions requises cible et d'importation d'IBM Cognos Real-time Monitoring.

Zones de message

Les zones de message XML sont définies dans le fichier de définition invokeExternalAction.xsd.

Ce fichier (ainsi que tous les fichiers XSD) se trouvent dans le répertoire suivant : `realtime\sdk\api\metadata`.

Pour plus d'informations sur les fichiers XML et XSD dans Cognos Real-time Monitoring, voir Chapitre 39, «Fichiers XML et XSD utilisés par IBM Cognos Real-time Monitoring», à la page 411.

Le fichier common.xsd situé dans le même répertoire est obligatoire.

Chaque message contient les zones suivantes :

- description

  Description du processus externe défini dans IBM Cognos Real-time Monitoring Workbench
- actionName

  Nom du processus externe défini dans Workbench.
- severity

  Gravité du message d'alerte d'origine ou sélectionnée par l'auteur du message.

Au besoin, d'autres zones sont insérées en fonction de l'objet affiché par l'utilisateur dans Cognos Real-time Monitoring Dashboard lors de l'initiation de l'action, par exemple l'objet d'un message d'alerte ou l'ensemble des lignes des données de la vue présentant un graphique. Pour de plus amples informations, consultez le fichier XSD.

Fichier WSDL de service Web

Lorsque vous implémentez un service Web, définissez-le pour recevoir un message SOAP, puis choisissez ses attributs.

Les attributs disponibles pour le service Web sont décrits dans le tableau ci-dessous :

Tableau 108. Attributs WSDL

Attribut	Valeur	Commentaire
Style	rpc	Ne pas utiliser de "document".
Espace-nom cible (tns)	http://cognos.obi.com	—
Encodage (soap:body)	encoded	Ne pas utiliser de "littéral".
Espace-nom d'importation	http://cognos.obi.com/2	Définit les types de données IBM Cognos Real-time Monitoring. Vous pouvez également les définir dans le fichier WSDL. Toutefois, cette définition n'est pas traitée dans le présent document.
Emplacement d'importation	Spécifique à l'installation	Emplacement du fichier invokeExternalAction.xsd dans votre installation.

Ci-après figure un exemple de fichier WSDL qui traite le message invokeExternalAction sur une machine et le port hôte 80. Notez qu'il importe le fichier de définition invokeExternalAction.xsd.

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:impl="http://cognos.obi.com/5"
```

```
xmlns:tns="http://cognos.obi.com"
    targetNamespace="http://cognos.obi.com"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
<import namespace="http://cognos.obi.com/5"
location="/cognos/obi/api/metadata/invokeExternalAction.xsd"/>
<message name="invokeExternalActionRequest">
    <part name="request" element="impl:invokeExternalAction"/>
</message>
<portType name="invokeExternalActionPortType">
    <operation name="invokeExternalActionOperation">
        <documentation>Receives a IBM Cognos Real-time
Monitoring external action.
        </documentation>
        <input message="tns:invokeExternalActionRequest"/>
    </operation>
</portType>
<binding name="invokeExternalActionBinding"
         type="tns:invokeExternalActionPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="rpc" />
    <operation name="invokeExternalActionOperation">
        <soap:operation soapAction=""/>
        <input>
            <soap:body
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
             namespace="http://cognos.obi.com" use="encoded"/>
        </input>
    </operation>
</binding>
<service name="invokeExternalActionService">
    <port name="invokeExternalActionService"
          binding="tns:invokeExternalActionBinding">
        <soap:address
location="http://host:80/axis/services/invokeExternalActionService"/>
    </port>
</service>
</definitions>
```

Chapitre 38. Fenêtres de requêtes

Une fenêtre de requêtes spécifie un ensemble de lignes utilisées dans des calculs par rapport à la ligne actuelle examinée, dans une vue créée avec une instruction C-SQL SELECT.

Pour plus d'informations sur les instructions C-SQL SELECT, voir Chapitre 32, «Instructions SELECT», à la page 337.

Les vues métier créées dans IBM Cognos Real-time Monitoring Workbench créent des instructions C-SQL en interne transmises aux serveurs. Le terme *événement* désigne une ligne ou un ensemble de lignes entrant simultanément dans le système. La source de ces lignes est un flux de données.

Présentation des fenêtres de requêtes

Dans une fenêtre, vous pouvez calculer une fonction d'ensembles mobiles (Moving set), une jointure ou bien supprimer des lignes dans une vue.

De tels calculs utilisent une fenêtre, mais dans les cas d'opérations par défaut, vous n'avez pas besoin de définir la sémantique de la fenêtre. Par exemple, en langage C-SQL, les fonctions d'ensembles effectuent des calculs sur les ensembles de lignes d'une vue. L'ensemble de lignes par défaut de chaque fonction correspond à tous les événements, compte tenu de l'initiation de la vue (ensemble illimité). Etudiez la vue suivante qui calcule la valeur totale d'une colonne nommée Qty correspondant à tous les événements insérés dans la vue :

```
SELECT SUM(Qty) AS Total_Qty
FROM Orders
```

Pour exprimer la fonction SUM() afin d'obtenir les mêmes résultats, vous pouvez également utiliser la syntaxe ci-dessous qui indique d'additionner l'ensemble de tous les événements précédents :

```
SELECT SUM(Qty)
OVER
(EVENTS UNBOUNDED PRECEDING) AS Total_Qty
    FROM Orders
```

La clause OVER définit une fenêtre qui détermine l'ensemble des lignes à inclure. Une fenêtre permet de limiter cet ensemble à un nombre spécifique de lignes d'événements ou aux événements survenus dans un laps de temps particulier. Pour totaliser uniquement les événements actuels et les cinq derniers événements, définissez une fenêtre comme suit :

```
SUM(Qty) OVER (EVENTS 5 PRECEDING)AS
Total_Qty
```

Et pour totaliser les événements du mois en cours, utilisez une fenêtre chronologique, comme suit :

```
SUM(Qty) (RANGE INTERVAL '1' MONTH PRECEDING)AS
Total_Qty OVER
```

© Copyright IBM Corp. 2007, 2014

Types de fenêtres

Toutes les fenêtres sont identifiées par la clause EVENTS ou RANGE et incluent une définition d'étendue qui définit leur taille.

```
{ RANGE | EVENTS }
<window frame extent >
```

La syntaxe d'étendue est propre au type de fenêtre et décrite en détail dans les sections «Fenêtres de série d'événements», à la page 394 et «Fenêtres chronologiques», à la page 396. Les autres clauses (non affichées) contrôlent le comportement de la fenêtre lorsque de nouveaux événements y sont entrés, les éléments à insérer ainsi que la méthode et le moment des mises à jour de la fenêtre pour inclure les nouveaux événements et supprimer les anciens.

Déclarations et références des fenêtres

Deux méthodes permettent de définir des fenêtres et de les associer à des fonctions.

En ligne

La déclaration en ligne définit les paramètres de la fenêtre qui suivent immédiatement la référence à la fonction.

Par exemple :

```
SELECT PartName, SUM(Qty) OVER (EVENTS 4
PRECEDING) AS Total_Qty,
   FROM Orders
   GROUP BY PartName
```

Ce format est utile si vous ne disposez que d'une fenêtre par requête même si vous pouvez également l'utiliser avec plusieurs fenêtres. Vous ne pouvez pas partager ces définitions de fenêtre entre les fonctions d'une même requête.

Référence par nom

Cette référence permet d'utiliser une fenêtre définie avec la clause WINDOW.

Par exemple :

```
SELECT PartName,
       SUM(Qty) OVER Previous4 AS
Total_Qty,
       AVG(Qty) OVER Previous4 AS
Average_Qty
   FROM Orders
   GROUP BY PartName
WINDOW Previous4 AS (EVENTS 4 PRECEDING)
```

Ce format est utile si vous disposez de plusieurs définitions de fenêtre simples, car vous pouvez les définir dans un seul emplacement, à savoir dans une même définition de clause WINDOW. Ce format vous permet également de partager la définition sur plusieurs fonctions d'une même requête (comme indiqué ci-dessus) et d'utiliser des fenêtres qui étendent la définition d'une autre fenêtre. Pour en savoir davantage, voir «Extension d'une définition de fenêtre avec une autre», à la page 393.

Les fonctions de la liste Select associée à une fenêtre doivent posséder des noms d'alias définis avec l'opérateur AS, comme AS Total_Qty dans les exemples ci-dessus.

Plusieurs fenêtres par requête

La clause WINDOW définit les fenêtres qui peuvent être partagées dans la requête et qui peuvent être étendues par d'autres fenêtres.

L'exemple suivant définit deux fenêtres, chacune étant utilisée par une fonction différente de la requête :

```
SELECT PartName,
       SUM(Qty) OVER Previous4 AS
Total_Qty_4,
       AVG(Qty) OVER Previous10 AS
Average_Qty_10
   FROM Orders
   GROUP BY PartName
WINDOW Previous4 AS (EVENTS 4 PRECEDING),
Previous10 AS (EVENTS 10 PRECEDING)
```

Extension d'une définition de fenêtre avec une autre

Lorsque des fenêtres partagent des traits communs, vous pouvez définir ces traits dans une définition de fenêtre, puis étendre (hériter) cette définition avec d'autres aspects uniques dans différentes fenêtres.

Par exemple, la définition suivante définit la fenêtre Common avec la clause PARTITION BY, puis d'autres fenêtres qui étendent les traits communs avec la portée requise pour les fenêtres uniques :

```
SELECT PartName,
       SUM(Qty) OVER Events4 AS
Total_Of_Qty_4,
       AVG(Qty) OVER Events10 AS
Average_Of_Qty_10
   FROM Orders
   WINDOW Common   AS (PARTITION
BY PartName),
          Events4  AS (Common EVENTS
4 PRECEDING),
          Events10 AS (Common EVENTS
10 PRECEDING)
```

La définition de fenêtre ci-dessus est identique à ce qui suit :

```
Events4  AS (PARTITION BY PartName EVENTS 4 PRECEDING),
Events10 AS (PARTITION BY PartName EVENTS 10 PRECEDING)
```

Restrictions

Les définitions d'extension de fenêtre ne peuvent pas inclure les propriétés définies dans la fenêtre de base.

Par exemple, vous ne pouvez pas définir de clause ORDER BY dans les fenêtres de base et d'extension. De plus :

- Une clause PARTITION BY peut s'afficher uniquement dans la définition de la fenêtre de base ; elle ne peut pas apparaître dans les extensions.

- Ces clauses peuvent s'afficher uniquement dans les définitions d'extension ; elles ne peuvent pas apparaître dans la fenêtre de base :

  BETWEEN/AND

  CURRENT EVENT

  EVENTS

  INITIALIZE

  RANGE

  REFERENCE

  SLIDE

  UNBOUNDED

  < window start integer > without an EVENTS or RANGE clause
- Une fenêtre ne peut étendre qu'une autre fenêtre définie dans la même requête ; une fenêtre d'une vue dérivée ne peut pas étendre une fenêtre d'une vue de base.

Fenêtres de série d'événements

Les fenêtres de série d'événements contiennent un ensemble défini d'événements maximal.

Initialement, la fenêtre est vide, mais elle est renseignée avec de nouveaux événements jusqu'à ce qu'elle atteigne sa capacité définie. Après quoi, les anciens événements sont supprimés un par un à mesure de l'insertion des nouveaux événements.

Figure 44. Fenêtres de série d'événements

Clause EVENTS

Les fenêtres comportant des séries d'événements sont identifiées par la clause EVENTS.

Il existe deux façons d'exprimer des plages d'événements. La première méthode consiste à utiliser les clauses BETWEEN et AND, qui définissent les limites supérieure et inférieure de la fenêtre. La seconde consiste à utiliser l'expression PRECEDING, qui indique uniquement la limite inférieure de la fenêtre. La limite supérieure représente l'événement en cours dans le second cas. L'expression PRECEDING précise que les lignes d'événements précèdent l'événement en cours.

```
([PARTITION BY < column >]
EVENTS  { BETWEEN {< oldestEvent>
| UNBOUNDED} PRECEDING
```

```
            AND { < newestEvent> PRECEDING | CURRENT
EVENT }
          |{< oldestEvent > | UNBOUNDED} PRECEDING
         }
 [SLIDE < distance >]
 [REFERENCE {FRAME | OPERATOR} ]
}
```

La clause PARTITION BY crée un cadre de fenêtre pour chaque <colonne> d'événement, à l'instar d'une fenêtre GROUP BY. Pour plus de détails, voir «Partitions des fenêtres», à la page 401.

La clause SLIDE détermine comment avancer la fenêtre lorsque de nouveaux événements sont insérés dans la vue. Pour plus de détails sur cette option, voir «Avancement des fenêtres», à la page 405.

La clause REFERENCE indique à la fenêtre à quel moment déterminer que des lignes ont expiré. Pour plus d'informations sur cette option, voir «Référence de mise à jour de fenêtre», à la page 408.

Exemples de clause Events

Cette fenêtre contient des lignes limitées à un nombre d'événements consécutifs dans la vue, par exemple les cinq derniers événements :

```
SUM(Qty) OVER (EVENTS 4 PRECEDING)AS
Total_Qty
```

ou bien les 10 événements commençant il y a 12 événements :

```
SUM(Qty) OVER (EVENTS BETWEEN 11 PRECEDING
AND 2 PRECEDING)
        AS Total_Qty
```

Remarquez que la taille du cadre de la fenêtre correspond à (<oldestEvent>-<newestEvent>+1), où <oldestEvent> représente l'événement le plus ancien et <newestEvent> l'événement le plus récent. Par exemple, le cadre ci-dessus contient 10 événements (11-2+1).

Pour inclure l'ensemble des événements précédents, y compris l'événement actuel, à l'aide de l'option UNBOUNDED, procédez comme suit :

```
SUM(Qty) OVER (EVENTS UNBOUNDED PRECEDING)AS
Total_Qty
```

Cette syntaxe se comporte comme si aucune fenêtre n'était définie :

```
SUM(Qty) AS Total_Qty
```

Evénement en cours

Le décompte se fait à partir de zéro. L'événement (0) représente l'événement en cours.

Dans les exemples de la section «Exemples de clause Events», notez que le numéro de l'événement de démarrage est inférieur de 1 à celui de l'événement de démarrage souhaité.

Figure 45. Méthode de calcul des événements

Voici une autre méthode permettant d'exprimer les cinq derniers événements :

```
OVER (EVENTS BETWEEN 4 PRECEDING AND 0 PRECEDING)
```

Une autre méthode permettant d'exprimer la plage d'événements consiste à utiliser le littéral CURRENT EVENT comme suit :

```
OVER (EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT )
```

Fenêtres chronologiques

Les fenêtres chronologiques s'agrandissent pour inclure tous les événements qui surviennent au cours d'une période.

Par exemple, la fenêtre 1-jour affichée dans l'illustration ci-dessous s'agrandit à l'arrivée de nouveaux événements au cours de la journée.

Figure 46. Fenêtre 1-jour

Ainsi, une fenêtre 3-jours inclut tous les événements de la fenêtre 3-jours actuelle.

Figure 47. Fenêtre 3-jours

Clause RANGE

Les fenêtres chronologiques sont identifiées par la clause RANGE.

Ces fenêtres contiennent des lignes limitées à une plage de dates et d'heures en combinant des années, des mois, des jours, des heures, des minutes ou des secondes. Les périodes peuvent être exprimées de deux façons différentes. La première méthode consiste à utiliser les clauses BETWEEN et AND, qui définissent les limites supérieure et inférieure de la fenêtre. (La limite supérieure est inclusive tandis que la limite inférieure est exclusive.) La seconde consiste à utiliser

l'expression PRECEDING, qui indique uniquement la limite inférieure de la fenêtre. La limite supérieure représente l'horodatage le plus récent dans le second cas.

```
([PARTITION BY < column >]
 [ORDER BY {< date-time column > | < integer column>}
[ ASC | DESC ] ]
RANGE {BETWEEN
           {INTERVAL < oldestTime > | < oldestInt>
| UNBOUNDED} PRECEDING
            AND {INTERVAL < newestTime > | < oldestInt>}
PRECEDING
        |{INTERVAL < oldestTime > | < oldestInt
> | UNBOUNDED} PRECEDING
       }
 [SLIDE [INTERVAL < distance >]]
 [REFERENCE {FRAME | OPERATOR} ]
 [INITIALIZE < initTimestamp >]
)
```

Lorsque vous faites appel à une clause RANGE, tenez compte des éléments suivants :

- Les colonnes "order-by", "oldest" et "newest" sont généralement des données de type date-heure. Cependant, vous pouvez également utiliser un nombre entier représentant une série chronologique. Pour en savoir davantage, voir «Séries chronologiques de nombres entiers», à la page 401.
- La clause PARTITION crée une fenêtre pour chaque < colonne > d'événements, à l'instar d'une fenêtre GROUP BY. Pour en savoir davantage, voir «Partitions des fenêtres», à la page 401.
- La clause ORDER BY identifie la colonne utilisée pour calculer l'heure de l'événement. Pour en savoir davantage, voir «Clause ORDER BY», à la page 399.
- La clause SLIDE détermine comment avancer la fenêtre lorsque de nouveaux événements sont insérés dans la vue. Pour en savoir davantage, voir «Avancement des fenêtres», à la page 405.
- INITIALIZE indique une date-heure commune à laquelle initialiser toutes les fenêtres associées d'une vue. Si vous utilisez une fenêtre chronologique, il est conseillé d'initialiser l'heure de démarrage à minuit pour les fenêtres de type jour, mois et année, au premier jour du mois pour les fenêtres de type mois et année et au premier jour de la période souhaitée pour les fenêtres de type année. Pour en savoir davantage, voir «Initialisation de fenêtre», à la page 409.
- La clause REFERENCE indique à la fenêtre à quel moment déterminer que des lignes ont expiré. Pour en savoir davantage, voir «Référence de mise à jour de fenêtre», à la page 408.
- ASC indique que les lignes sont classées en fonction de leur horodatage, du plus ancien au plus récent. Pour plus d'informations sur l'ordre de tri, voir «Ordre décroissant», à la page 400.
- DESC indique que les lignes sont classées en fonction de leur horodatage, du plus récent au plus ancien. Pour plus d'informations sur l'ordre de tri, voir «Ordre décroissant», à la page 400.

Exemples de clause RANGE

Cette fenêtre totalise les événements du mois en cours de la colonne Qty.

```
SUM(Qty) OVER (RANGE INTERVAL '1' MONTH PRECEDING)AS
Total_Qty
```

Vous pouvez également identifier des périodes très spécifiques, par exemple celle qui commence il y a 18 heures et 15 minutes et se termine il y a 45 secondes. Pour en savoir plus sur les spécifications date-heure, voir «Date-heure», à la page 76.

```
OVER (RANGE BETWEEN INTERVAL '18:15' HOUR TO MINUTE PRECEDING
      AND INTERVAL '45' SECOND PRECEDING
)
```

Quels sont les événements inclus ?

Les événements à inclure dans une fenêtre chronologique sont déterminés lorsqu'un nouvel événement est entré dans la fenêtre ou la vue.

Pour en savoir davantage, voir «Référence de mise à jour de fenêtre», à la page 408.

L'intervalle inclut les événements qui correspondent à la taille de la période de l'événement en cours. Par exemple, consultez la fenêtre ci-dessous dont l'intervalle est de 1 jour :

```
SUM(order_qty) OVER ( RANGE INTERVAL '1' DAY  PRECEDING
AS TotalQty
```

Si deux événements sont séparés d'un jour, ils sont tous les deux inclus dans la fenêtre. Notez la valeur de la colonne TotalQty après les deux événements :

```
order_qty TotalQty Event Time
--------- -------- ---------------------
        1        1 2003-12-01 09:00:00.0
        1        2 2003-12-02
09:00:00.0
```

Lors de l'utilisation de la clause BETWEEN, la valeur <oldestTime> est exclue tandis que la valeur <newestTime> est incluse. Par exemple, pour disposer de deux fenêtres, l'une de la semaine en cours et l'autre de la semaine précédente, utilisez ces définitions :

```
ThisWeek:  RANGE INTERVAL '7'
DAY PRECEDING
LastWeek:  RANGE BETWEEN INTERVAL '14' DAY PRECEDING
                 AND INTERVAL '7'
DAY PRECEDING
```

Notez que ces deux plages utilisent 7 comme valeur associée. La semaine en cours inclut tous les événements qui se sont produits entre le jour en cours et il y a sept jours alors que la semaine précédente inclut les sept jours avant la semaine en cours. Une autre façon de définir les fenêtres ci-dessus consiste à utiliser "0" comme heure actuelle, par exemple :

```
ThisWeek:  RANGE BETWEEN INTERVAL '7' DAY PRECEDING
                 AND INTERVAL '0'
DAY PRECEDING
LastWeek:  RANGE BETWEEN INTERVAL '14' DAY PRECEDING
                 AND INTERVAL '7' DAY PRECEDING
```

La ligne correspondant précisément à il y a sept jours n'est pas incluse dans cette semaine, mais dans la semaine précédente.

Clause ORDER BY

Sauf définition contraire, toutes les heures sont calculées en fonction de l'horodatage d'arrivée interne de chaque événement.

Par exemple, cette fenêtre contient les événements de la dernière heure dans l'ordre de leur arrivée sur le système.

```
OVER (RANGE INTERVAL '1' HOUR PRECEDING)
```

Vous pouvez désigner une colonne de type date-heure comme étant la colonne principale de l'événement. Par exemple, vous pouvez utiliser l'heure à laquelle une commande a été passée :

```
OVER (ORDER BY order.order_timestamp RANGE
'4' DAY PRECEDING)
```

L'argument ORDER BY est une référence de nom de colonne unique ; vous ne pouvez pas utiliser de nombres entiers pour représenter la position ordinale de la colonne dans la liste SELECT. Par ailleurs, l'ordre de tri par défaut est croissant (ASC), même si l'ordre décroissant (DESC) est également disponible. Pour plus de détails, voir «Ordre décroissant», à la page 400. La clause ORDER BY respecte la syntaxe suivante :

```
ORDER BY < columnNameReference > [ ASC | DESC ]
```

Arrivée dans n'importe quel ordre

Lorsque le flux d'événements ne respecte pas l'ordre attendu, le moteur d'interrogation tente d'insérer l'événement désordonné à son emplacement correct dans le cadre de la fenêtre et met à jour en conséquence toutes les agrégations.

Le moteur d'interrogation utilise toujours la dernière heure de réception de tous les événements avant l'événement désordonné pour déterminer s'il est inclus. Si l'événement ne figure pas dans le dernier cadre de la fenêtre, il est omis.

Prenez l'exemple d'une fenêtre 1 heure qui reçoit les événements suivants dans l'ordre indiqué. Dans cet exemple, le deuxième événement est le dernier, et ainsi, seuls les événements reçus après 08:10 (09:10 moins 1 heure) sont inclus dans la fenêtre. Le quatrième événement est donc rejeté puisque son horodatage indique 08:04.

```
OVER (ORDER BY Time RANGE INTERVAL
'1' HOUR PRECEDING)
Arrival
Sequence  Time   Amount
--------  -----  ------
 1        08:45  242.69
 2        09:10  103.76  << Latest timestamp sets
the window
 3        08:50   90.20  << Out-of-order, accepted
 4        08:04  188.88  << Out-of-order, rejected
```

Une fois que les quatre événements ci-dessus ont été traités, la vue qui les contient apparaît ainsi :

```
Arrival
Sequence  Time   Amount
--------  -----  ------
 1        08:45  242.69
```

```
 3        08:50   90.20  << Reordered
 2        09:10  103.76
```

Ordre décroissant

Par défaut, l'ordre est supposé croissant : les anciens événements sont traités en premier.

Cependant, vous pouvez spécifier DESC pour indiquer l'ordre de tri décroissant. Lorsque des événements arrivent et qu'ils ne sont pas déjà dans l'ordre décroissant, ils sont traités de la même façon que dans la section «Arrivée dans n'importe quel ordre», à la page 399.

```
OVER (ORDER BY Time DESC RANGE
INTERVAL '1' HOUR PRECEDING)
Arrival
Sequence  Time   Amount
--------  -----  ------
 1        08:45  242.69
 2         09:10  103.76 << Out-of-order, accepted,
expires after fourth event
 3        08:50   90.20 << Out-of-order, accepted
 4        08:04  188.88
```

Une fois que les quatre événements ci-dessus ont été traités, la vue qui les contient apparaît ainsi :

```
Arrival
Sequence  Time   Amount
--------  -----  ------
 4        08:04  188.88
 1        08:45  242.69
 3        08:50   90.20
```

Horodatages de valeur NULL

Lorsque la colonne référencée contient une valeur NULL correspondant à l'horodatage, l'événement est rejeté à moins que la plage ne soit ILLIMITEE.

Lorsque le cadre de la fenêtre est illimité, tous les événements sont inclus, y compris les horodatages NULL, bien que les valeurs NULL soient placées en dernier dans l'ensemble, suivant leur ordre d'arrivée. Par exemple :

```
Arrival
Sequence  Time   Amount
--------  -----  ------
 1        08:04  188.88
 2        08:45  242.69
 3               103.76  << First NULL time
 4        08:50   90.20
 5               157.11  << Second NULL time
```

Lorsque les événements ci-dessus sont traités dans l'ordre croissant en fonction de la colonne Time, la vue finale ressemble à celle-ci :

```
Arrival
Sequence  Time   Amount
```

```
--------  -----  ------
 1        08:04  188.88
 2        08:45  242.69
 4        08:50   90.20
 5                157.11  << Second NULL time
 3                103.76  << First NULL time
```

Séries chronologiques de nombres entiers

Une plage chronologique est exprimée sous la forme de valeurs de type date-heure ou de valeurs d'intervalle de dates.

Cependant, vous pouvez également utiliser un nombre entier représentant une série chronologique. Par exemple, examinez cette série de valeurs de type date-heure et leurs correspondances en nombres entiers :

```
Date_time_value       Date_time_int
-------------------   -------------
2003-12-01 09:00:00      3795637500
2003-12-01 13:00:00      3795654167
2003-12-02 09:00:00      3795737500
2003-12-03 09:01:00      3795837569
2003-12-04 09:00:00      3795937500
```

L'heure représentée par les nombres entiers est tout à fait arbitraire et n'est pas un facteur du mode de traitement des valeurs par IBM Cognos Real-time Monitoring. Vous devez connaître la signification des valeurs. Dans l'exemple de série ci-dessus, 100 000 correspond à un jour. Une fenêtre des deux jours précédents correspondant à ces données est définie ainsi :

```
ORDER BY Date_time_int
RANGE 200000 PRECEDING
```

Ou celle permettant de voir uniquement le jour précédent est définie ainsi :

```
ORDER BY Date_time_int
RANGE BETWEEN 200000 PRECEDING AND 100000 PRECEDING
```

Lors de l'utilisation d'une série chronologique de nombres entiers, utilisez toujours la clause ORDER BY et identifiez la colonne de nombres entiers lors du classement de la série.

Partitions des fenêtres

Tous les événements d'une vue sont insérés dans un même ensemble (fenêtre) sauf définition contraire par la clause GROUP BY ou PARTITION BY.

Ces clauses trient les événements et les répartissent dans des fenêtres en fonction d'une valeur clé, par exemple un nom ou un ID commun. Les fenêtres partitionnées permettent d'agréger les événements propres à la fenêtre. Par exemple, pour collecter le volume total de l'ensemble des transactions en fonction de la sécurité, la définition de la vue pourrait être ainsi :

```
SELECT Trades.symbol, SUM(Trades.volume) OVER Symbols
AS Total_volume
   FROM Trades
   WINDOW Symbols AS ( PARTITION BY Trades.symbol )
```

Cette vue est semblable à celle définie avec la clause GROUP BY de la manière suivante :

```
SELECT Trades.symbol, SUM(Trades.volume) AS Total_volume
   FROM Trades
GROUP BY Trades.symbol
```

L'interrogation des deux vues ci-dessus produit des résultats différents. La vue groupée renvoie une ligne par groupe alors que la vue partitionnée, par défaut, renvoie uniquement une ligne contenant les résultats du dernier événement entré dans la vue (même si les informations de chaque partition sont conservées en interne). Pour afficher plus de lignes dans une vue partitionnée, définissez le paramètre **Garder dans la vue** sur une taille supérieure à 1. Pour plus d'informations, voir «Résultats historiques des vues partitionnées», à la page 404.

Clause PARTITION BY

La clause PARTITION BY définit une ou plusieurs colonnes qui contiennent les valeurs identifiant une fenêtre de partition.

```
PARTITION BY < column > [, < column > ... ]
```

Où <column> correspond au nom d'une colonne dans la liste SELECT ou bien à un nombre entier ordinal qui représente la position d'une colonne répertoriée dans la liste SELECT des colonnes (la première référence est 1). Lorsque vous déclarez une liste de colonnes, une partition est créée pour chaque valeur unique de l'ensemble. Par exemple, cette déclaration crée des partitions pour des particuliers en fonction des combinaisons nom+prénom :

```
PARTITION BY last_name,first_name
```

Examinez les illustrations des sections suivantes afin de voir ce qui se passe lors de l'arrivée d'un nouvel événement plus récente que l'événement le plus récent déjà dans la vue.

Mise à jour des vues pour une clause GROUP BY simple

Lorsqu'un événement arrive dans une vue avec une clause simple GROUP BY, il est appliqué au groupe associé.

Le cas échéant, ce nouvel événement est appliqué à la moyenne de tous les événements AAA reçus :

Figure 48. Mise à jour des vues pour une clause GROUP BY simple

Mise à jour des vues pour une partition avec référence de cadre

Maintenant prenez l'exemple du même événement entrant dans une vue partitionnée par nom et où seul le cadre de la fenêtre de partition qui identifie l'événement est mis à jour.

Dans cet exemple, l'événement AAA est postérieur de 2 heures au dernier événement AAA. Ainsi, toutes les valeurs précédentes de la partition expirent et sont supprimées, et seul le nouvel événement est utilisé. Notez que l'autre partition n'est pas affectée.

Figure 49. Mise à jour des vues pour une partition avec référence de cadre

Mise à jour des vues pour une partition avec référence d'opérateur

Pour finir, prenez l'exemple de ce même événement arrivant dans une vue partitionnée qui est mise à jour en fonction de REFERENCE OPERATOR.

La référence indique à toutes les partitions de se mettre à jour à l'arrivée d'un nouvel événement dans la fenêtre. Dans cette illustration, toutes les partitions existantes expirent et une nouvelle partition est créée pour le nouvel événement, car aucun des événements suivis par les partitions existantes n'est compris dans la plage de la dernière heure.

Figure 50. Mise à jour des vues pour une partition avec référence d'opérateur

La requête suivante est quasi identique à celle de l'illustration précédente, sauf que dans l'exemple précédent, seule la ligne correspondant au dernier événement entré dans la vue est affichée dans le résultat :

```
SELECT Name, MOV_AVG(Value,HOUR,1)GROUP BY
Name
```

Avantage des partitions sur les groupes

Le fait de pouvoir disposer de plusieurs partitions basées sur les différentes colonnes d'une même vue alors que la clause GROUP BY s'applique exclusivement à toute la vue constitue le principal avantage des partitions.

Utilisation de la clause GROUP BY dans les fenêtres sur le point d'expirer

Une clause GROUP BY permet d'afficher les résultats de tous les groupes dans IBM Cognos Real-time Monitoring Dashboard ou dans l'onglet **Résultats** de Workbench.

Il s'agit d'un avantage sur les partitions, qui ne vous permettent pas de consulter de vue pour y observer le contenu des partitions.

Une clause GROUP BY conserve les résultats de chaque groupe tant que le groupe contient des données. Dans l'illustration précédente, si vous utilisez la vue créée avec l'expression GROUP BY au lieu de PARTITION, vous pouvez afficher le contenu, et les groupes expirent s'ils ne contiennent plus d'événements de la dernière heure.

Résultats historiques des vues partitionnées

Lorsque vous interrogez une vue partitionnée, par défaut, le résultat correspond à une vue pourvue d'une ligne contenant le résultat du dernier événement entré dans la vue (même si les informations de chaque partition sont conservées en interne).

Par exemple, si vous effectuez le suivi du prix moyen de titres de placement, partitionné par symbole, l'interrogation de SELECT * sur la vue renvoie un résultat semblable à ce qui suit :

```
SELECT * FROM AveragePricesView
Symbol  AvgPrice  Date
------  --------  ----------
JMH     164.35    2003-07-14
```

Dans les résultats ci-dessus, le dernier événement reçu par la vue AveragePricesView correspond au symbole JMH.

Pour afficher plus de lignes, définissez le paramètre **Garder dans la vue** de cette vue partitionnée sur une taille supérieure à 1. Si vous disposez de nombreux événements agrégés, attribuez-lui une valeur supérieure, par exemple 500. Ainsi, l'interrogation de la vue renvoie de nombreuses lignes, comme indiqué dans l'exemple suivant :

```
Symbol  AvgPrice  Date
------  --------  ----------
AAA      24.35    2003-03-05
SRKH    102.07    2003-03-05
JMH      90.22    2003-03-05
SRKH    106.88    2003-03-06
AAA      25.66    2003-03-06
JMH      94.11    2003-03-06
...
```

Les résultats apparaissent dans l'ordre dans lequel ils ont été générés par la vue : ordre des derniers événements entrés dans chaque partition.

Pour obtenir des résultats à signification historique, triez la nouvelle vue en fonction des colonnes Symbol (Symbole) et Date. Par exemple :

```
SELECT * FROM AveragePricesView ORDER BY Symbol,"Date"
Symbol  AvgPrice  Date
------  --------  ----------
AAA      24.35    2003-03-05
AAA      25.66    2003-03-06
AAA      25.25    2003-03-07
AAA      24.92    2003-03-08
...
```

Avancement des fenêtres

Lorsqu'un nouvel événement entre dans une fenêtre, celle-ci détermine les événements à conserver et à supprimer lorsqu'elle est pleine.

Un cadre de fenêtre est plein si la ligne suivante provoque l'expiration d'une ligne existante de la fenêtre. Lorsqu'elle est affichée dans le contexte des événements prochains et passés du flux d'événements, la fenêtre peut être vue en avancement ou en glissement le long du flux d'événements à mesure de l'ajout et de la suppression d'événements.

Figure 51. Avancement des fenêtres

Clause SLIDE

La clause d'avancement de fenêtre (SLIDE) spécifie la distance d'avancement lorsque la fenêtre est pleine.

Par défaut, lorsque la clause SLIDE est omise, les fenêtres de série d'événements glissent d'un événement à la fois le long du flux à l'arrivée de chaque nouvel événement, alors que les fenêtres chronologiques s'avancent pour inclure le dernier événement, et tous les événements dans l'intervalle défini dans la clause RANGE sont conservés, les autres événements étant supprimés.

L'insertion d'une clause SLIDE fait avancer la fenêtre de toute sa taille (si vous omettez l'argument) ou bien la fait avancer d'un événement ou de l'intervalle de temps spécifié par l'argument.

```
SLIDE [ < interval > | < distance > ]
```

Prenez l'exemple d'une fenêtre de série d'événements dont la taille représente trois événements. La déclaration d'une clause SLIDE sans argument a le même effet qu'une déclaration SLIDE 3.

Figure 52. Fenêtre de série d'événements dont la taille représente trois événements

Lorsqu'une fenêtre avance de plusieurs événements, elle est dite en cascade. les fonctions TUMBLE_ constituent un raccourci des expressions de fenêtre complètes utilisant ce comportement de glissement. Pour en savoir davantage, voir «Fonctions tumble», à la page 407.

L'argument SLIDE ne doit pas être identique à la taille de la fenêtre, mais il doit être inférieur ou égal à la taille de la fenêtre.

Fenêtres en défilement continu

Une fenêtre en défilement continu se vide de son contenu lorsqu'elle avance pour inclure le tout dernier événement.

Par exemple, une fenêtre chronologique en défilement continu s'agrandit jusqu'à ce que l'arrivée d'un nouvel événement provoque la suppression des anciens événements. Si l'intervalle SLIDE est identique à la taille de la fenêtre, la fenêtre vide tous les événements existants à l'arrivée d'un nouvel événement et lorsqu'elle est pleine.

Figure 53. Fenêtres en défilement continu

Cette fonctionnalité permet d'effectuer le suivi des événements de l'intervalle complet pendant cet intervalle. Par exemple, si vous commencez un intervalle un dimanche et déclarez un intervalle de glissement de sept jours, la fenêtre se vide et s'avance tous les dimanches. Utilisez la clause INITIALIZE pour définir l'heure de début de façon appropriée. Pour en savoir davantage, voir «Initialisation de fenêtre», à la page 409.

Fenêtres Défilement continu de fin

Une fenêtre Défilement continu de fin se vide, puis commence à se remplir après son glissement.

Cependant, il importe de comprendre que les éléments qui entrent dans la vue doivent être contenus dans la fenêtre identifiée par le dernier événement. Prenez l'exemple de cette fenêtre sept jours qui contient des valeurs de la semaine précédente et se vide tous les sept jours :

```
RANGE BETWEEN INTERVAL '14' DAY PRECEDING
      AND INTERVAL '7' DAY PRECEDING
SLIDE INTERVAL '7' DAY
```

Cette fenêtre accepte uniquement les valeurs antérieures aux sept jours précédant le dernier événement. Même s'il peut y avoir des événements d'une semaine complète, cette fenêtre ne contient que les événements antérieurs de sept jours au dernier événement. Par exemple, si ces événements sont entrés dans la fenêtre, seul le premier y est inséré, car son antériorité au dernier événement est supérieure à sept jours :

```
Data_Stream_Time
-------------------
2003-12-01 09:00:00   << Only event included
in the window
2003-12-01 10:00:00
2003-12-02 08:45:00
2003-12-08 09:10:00   << Last event
```

Les trois premiers événements seront inclus uniquement après la réception d'un événement le 09/12 après 08:45.

Fonctions tumble

La plupart des fonctions d'ensembles (Set) disposent de fonctions tumble_ associées, qui constituent un raccourci des expressions complètes de glissement de fenêtre.

Prenez par exemple cette expression SUM() en cascade qui additionne tous les événements se produisant dans un intervalle de 1 heure :

```
TUMBLE_SUM(price, HOUR, 1, trade_time) AS T_Sum
```

La fonction ci-dessus est un raccourci de l'expression de la fenêtre en ligne suivante :

```
SUM(price) OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE) AS T_Sum
```

Qui à son tour équivaut à l'expression suivante après avoir renseigné toutes les valeurs par défaut :

```
SUM(price) OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR) AS T_Sum
```

De même, la fonction TUMBLE_SUM(price, EVENT, 5) est un raccourci de cette fenêtre complète :

```
SUM(price) OVER (
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE FRAME) AS T_Sum
```

Pour plus de détails sur le comportement des fonctions en cascade particulières, consultez leurs descriptions.

Référence de mise à jour de fenêtre

Lorsqu'une vue reçoit un nouvel événement, une ou plusieurs fenêtres de la vue peuvent être mises à jour pour refléter les nouvelles informations.

En fonction de la référence et de la définition de chaque fenêtre, il est possible que tous les événements d'une fenêtre expirent et soient supprimés de la fenêtre ou ne soient pas affectés par la mise à jour.

Les événements filtrés avant d'entrer dans la vue, par exemple lorsqu'ils sont exclus par une clause WHERE, n'affectent pas les fenêtres de la vue et ne provoquent pas leur mise à jour, quel que soit le point de référence.

Pour plus d'informations sur la clause WHERE, voir «Clause WHERE», à la page 343.

Clause REFERENCE

Une référence détermine le moment auquel évaluer l'ensemble des événements inclus dans une fenêtre.

Les références sont les suivantes :

- OPERATOR

  La fenêtre est mise à jour à chaque entrée d'un nouvel événement dans la vue, que cet événement soit ou non inclus dans la fenêtre. Par exemple, si une vue effectue le suivi des titres de placement échangés au cours de la dernière heure et partitionne chaque titre dans sa propre fenêtre, chaque fenêtre évalue son ensemble à chaque entrée d'un nouvel échange dans la vue. Si un titre n'a pas été échangé au cours de la dernière heure, sa fenêtre se vide. Toutes les autres fenêtres incluent uniquement les titres échangés au cours de la dernière heure ; les anciens échanges sont supprimés de leurs fenêtres.

  ```
  SELECT
  Trades.symbol,
         AVG(Trades.price)
            OVER (RANGE INTERVAL '1' HOUR PRECEDING REFERENCE
  OPERATOR)
            AS av_price_last_hour
     GROUP BY symbol
     FROM Trades
  ```

  Ce mode est souhaitable (mode par défaut) si vous utilisez des agrégations chronologiques et souhaitez que toutes les fenêtres incluent des événements référencés à partir de la même heure : heure d'arrivée du dernier événement dans la vue. Ce mode est également utile aux vues dans lesquelles vous voulez que les fenêtres de séries d'événements expirent et réduisent la consommation de la mémoire par la vue.

- FRAME

La fenêtre est mise à jour uniquement lorsqu'un nouvel événement y entre. Par exemple, si la vue effectue le suivi des 10 derniers échanges de chaque titre de placement, seule la fenêtre recevant le dernier échange est mise à jour. Toutes les autres fenêtres (titre) conservent leurs 10 événements.

```
SELECT Trades.symbol,
       AVG(Trades.price) OVER (EVENTS 9 PRECEDING REFERENCE
FRAME)
          AS av_price_last_10_trades
   GROUP BY symbol
   FROM Trades
```

Ce mode est souhaitable si vous souhaitez que les fenêtres conservent un ensemble d'événements, quelle que soit leur heure/date d'arrivée, notamment pour les agrégats mobiles basés sur les événements. Il s'agit du mode par défaut des événements basés sur les fenêtres.

La référence FRAME est également utile pour récupérer de la mémoire sur le serveur.

Initialisation de fenêtre

Lors de l'utilisation d'une fenêtre chronologique, l'heure de début du cadre de la fenêtre est définie par le premier événement arrivant dans la fenêtre.

Lorsqu'une vue contient plusieurs cadres de groupe ou de partition, chacun peut avoir une heure de début différente. Prenez ces deux événements qui sont les premiers à arriver dans la vue :

```
Symbol  Time
IBM     09:00:00.875
CQST    09:23:02.111
```

Si la vue qui reçoit ces événements les place dans différents cadres GROUP BY, chacun démarre à l'heure de chaque événement et est toujours réinitialisé en fonction de cette heure d'initialisation :

```
SELECT Trades.Symbol, Trades.Time,
       AVG(Trades.Price) OVER An_Hour AS Avg_Price_One_Hour_Tumble
   GROUP BY Trades.Symbol, Trades.Time
   FROM Trades
   WINDOW An_Hour AS (ORDER BY Trades.Time
                     RANGE INTERVAL '1' HOUR PRECEDING
SLIDE)
```

Avec cette définition de vue, un événement arrivant à 09:10 provoque l'expiration de l'événement IBM, mais l'événement CQST est conservé dans sa fenêtre pendant au moins 13:02 minutes : il s'agit du temps restant depuis son entrée dans la vue.

Clause INITIALIZE

Pour que toutes les fenêtres commencent à la même heure, utilisez la clause INITIALIZE.

Celle-ci définit le point d'initialisation pour tous les cadres basés sur la définition de fenêtre. Par exemple, pour que toutes les fenêtres commencent à la même heure, initialisez-les sur une date-heure antérieure au premier événement susceptible d'arriver dans la vue.

```
   WINDOW An_Hour AS (ORDER BY Trades.Time
```

```
                    RANGE INTERVAL '1' HOUR PRECEDING
SLIDE
INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000')
```

Avec cette définition, toutes les fenêtres sont initialisées à la même heure : minuit. Ainsi, chaque cadre expire au tout début de l'heure (lorsque les minutes correspondent à 00:00.000). Comme la fenêtre inclut la clause SLIDE, tous les échanges précédents sont supprimés à l'expiration du cadre, et seuls les nouveaux événements arrivant pendant l'heure en cours sont acceptés.

Cette clause sert de filtre et exclut tous les événements avant l'heure d'initialisation.

L'heure d'initialisation est une valeur de littéral de type date-heure. Pour en savoir davantage, voir «Littéral TIMESTAMP», à la page 78. Par ailleurs, la valeur de l'initialisation est statique : elle ne peut pas être modifiée après la création de la vue.

Exemple de clause INITIALIZE

Cette définition d'initialisation indique l'exercice fiscal en cours, qui débute le 1er juillet de l'année calendaire.

```
(RANGE INTERVAL '1' YEAR PRECEDING SLIDE
 INITIALIZE TIMESTAMP '1963-07-01 00:00:00')
```

Chapitre 39. Fichiers XML et XSD utilisés par IBM Cognos Real-time Monitoring

Tous les objets IBM Cognos Real-time Monitoring et de nombreuses opérations système peuvent être exprimés au format XML et téléchargés sur le système.

La syntaxe XML doit être correctement mise en forme conformément à la définition des fichiers de schéma XML (XSD). Pour obtenir la liste complète des objets définis et des opérations que vous pouvez exécuter, voir «Fichiers XSD Cognos Real-time Monitoring», à la page 418.

Les détails sur le format XML et les fichiers XSD ne sont pas traités dans cette documentation. Pour plus d'informations sur XML et XSD, voir http://www.xml.org.

Fichiers XML et XSD dans Cognos Real-time Monitoring

Tous les fichiers XML possèdent des fichiers de schéma qui définissent la structure du langage XML.

Les fichiers XSD et les fichiers d'exemple XML sont disponibles dans le répertoire *emplacement d'installation*/realtime/sdk/api/metadata/, où *emplacement d'installation* est le répertoire dans lequel IBM Cognos Business Intelligence a été installé.

Outre les exemples, une autre façon de voir des fichiers XML correctement définis consiste à créer d'abord des objets dans Cognos Real-time Monitoring Workbench ou Cognos Real-time Monitoring Dashboard, puis à utiliser la Console d'administration pour exporter tout l'ensemble. Tous les objets exportés sont écrits sous forme de fichiers XML dans le répertoire d'exportation du serveur. Pour plus d'informations sur cette procédure, voir la section sur l'importation et l'exportation des métadonnées du *Guide d'utilisation d'IBM Cognos Real-time Monitoring Workbench*.

Dépendances

La plupart des objets du système dépendent d'autres objets.

Lorsque vous définissez un nouvel objet, toutes ses dépendances doivent d'abord être définies. Pour ce faire, définissez et téléchargez les objets de base dans leur ordre de dépendance ou définissez-les en lot conformément à la description de la section «Définition de plusieurs objets avec le langage XML», à la page 415.

Espace

Lorsqu'une valeur d'élément XML contient plusieurs caractères d'espace contigus qui doivent être conservés, indiquez au système de les conserver avec la syntaxe suivante : xml:space="preserve".

Dans le cas contraire, la spécification XML indique de supprimer les espaces supplémentaires. Par exemple, sans la directive de conservation, la description suivante serait tronquée des espaces de fin et ne présenterait qu'un espace entre les mots :

© Copyright IBM Corp. 2007, 2014

```
<description xml:space="preserve">A
  note   </description>
```

Caractères d'échappement

Les caractères "<" et "&" sont incorrects au format XML.

D'autres caractères sont autorisés mais peuvent être source de confusion. Pour ces derniers, utilisez plutôt ces entités d'échappement XML.

Tableau 109. Entités d'échappement XML

Séquence	Caractère	Résultat
<	<	Inférieur à
>	>	Supérieur à
&	&	Perluète
'	'	Apostrophe
"	"	Guillemet

Par exemple, pour exprimer une requête contenant un symbole Inférieur à, utilisez plutôt &lt, comme indiqué dans l'exemple ci-dessous :

```
<query>SELECT c1, c2 FROM event1 WHERE c3&lt;=100
AND c2=&apos;CQST&apos;</query>
```

Données de type caractères

Au lieu d'utiliser des caractères d'échappement, vous pouvez utiliser une balise CDATA pour exprimer des caractères spéciaux.

Cette balise indique à l'analyseur d'ignorer tous les caractères spéciaux et de les traiter comme des littéraux.

```
<query><![CDATA[SELECT
c1, c2 FROM event1
WHERE c3<=100 AND c2='CQST']></query>
```

Téléchargement de fichiers XML

Deux méthodes permettent de télécharger des fichiers XML dans IBM Cognos Real-time Monitoring.

Ces deux méthodes sont les suivantes :

- Téléchargement de fichiers XML à partir d'une ligne de commande avec l'utilitaire cqupload.jar. Pour en savoir davantage, voir «Téléchargement de fichiers XML à partir d'une ligne de commande», à la page 413.
- Téléchargement de fichiers XML à partir d'un navigateur Web avec le script fileupload.jsp. Pour en savoir davantage, voir «Téléchargement de fichiers XML à partir d'un navigateur Web», à la page 413.

Ces deux méthodes nécessitent que le serveur Cognos Real-time Monitoring soit en cours d'exécution. Chaque méthode utilise également un compte utilisateur Cognos Real-time Monitoring pour se connecter au serveur et exécuter l'action. Dans chaque cas, le compte doit disposer de droits de création sur la classe d'objets à créer ou de droits d'administration sur les opérations à exécuter. Pour plus d'informations sur les comptes utilisateur, voir Chapitre 34, «Utilisateurs», à la page 355.

Téléchargement de fichiers XML à partir d'une ligne de commande

A partir d'une ligne de commande, exécutez l'utilitaire cqupload.jar pour télécharger des fichiers.

L'utilitaire est disponible à l'emplacement suivant : `realtime\sdk\java\lib`.

Cet utilitaire dispose de trois options :

Tableau 110. Options de l'utilitaire de téléchargement

Option	Description
-s applicationURL	Adresse URL qui localise Cognos Real-time Monitoring.
-u userName	Compte utilisateur à utiliser. Omettez cette option pour utiliser le compte d'administrateur système par défaut.
-p password	Mot de passe du compte utilisateur. Il est requis si vous insérez -u.

Par exemple, pour télécharger un fichier XML à l'aide du compte d'administrateur système par défaut :

```
java -jar /CognosLava/CQupload/cquload.jar
```

```
java -jar <CD-ROM>/LavaActivitySuite/CQUpload/cqupload.jar
-s http://applicationServer >/now createUserSkyler.xml
```

Pour inclure un nom d'utilisateur ou un mot de passe :

```
java -jar cqupload.jar -u skyler -p roo -s
...
```

Une opération réussie se produit automatiquement. Si elle échoue, l'utilitaire renvoie un message d'erreur à la fenêtre de commande. Passez en revue ce message afin d'identifier le problème. Par exemple, ce message indique une erreur dans la syntaxe XML :

```
Error uploading file: createUserSkyler.xml
---------------------------------------------------------------------
java.io.IOException: Error parsing an XML document. Ensure
that the
XML conforms exactly to the XML schema definition. The
XML that
cannot be parsed is:
```

Vous trouvez la cause effective de l'échec plus bas dans le message : un élément non valide.

```
Caused by:
javax.xml.bind.UnmarshalException: Unexpected element
{http://cognos.obi.com/5}:nome
```

Téléchargement de fichiers XML à partir d'un navigateur Web

Le script `fileupload.jsp` présente un formulaire dans lequel vous identifiez le fichier XML à télécharger.

Le fichier `fileupload.jsp` inclut un dispositif de sélection de fichier dans lequel vous identifiez le fichier XML à charger et affiche un message avec les résultats du téléchargement.

Procédure

1. Exécutez le script fileupload.jsp. Utilisez une adresse semblable à l'URL ci-dessous. Utilisez le système hôte local uniquement si vous exécutez le navigateur sur le même hôte que les serveurs IBM Cognos Real-time Monitoring. Dans le cas contraire, utilisez le même emplacement que celui que vous utilisez pour exécuter Cognos Real-time Monitoring Workbench.

   ```
   http://localhost/now/jsp/fileupload.jsp
   ```

2. Identifiez le nom d'utilisateur et le mot de passe à utiliser.
3. Identifiez le fichier XML à télécharger.
4. Sélectionnez **Télécharger**.

   La page de résultats affiche le nom du fichier XML téléchargé ainsi que ses résultats. Si l'état est Echec, passez en revue l'exception pour connaître l'origine du problème, puis corrigez le problème. Par exemple, ce message indique que l'opération a échoué, car l'objet utilisateur Rolf n'existe pas.

   ```
   Exception Encountered
   com.cognos.obi.exception.VCException: Cannot alter the
   [User]
   named [Rolf] because that object does not exist.
   ```

Définition d'un objet avec le langage XML

Cette section explique comment définir et télécharger une définition XML.

Pour télécharger plusieurs fichiers XML dans des objets spécifiques avec dépendances, suivez les instructions de la section «Définition de plusieurs objets avec le langage XML», à la page 415.

Procédure

1. Créez une définition XML. Utilisez le fichier XSD associé pour déterminer les éléments valides du fichier XML. Pour consulter un exemple de fichier complet, voir «Exemple : Création d'un utilisateur».
2. Assurez-vous de l'exécution de Cognos Real-time Monitoring.
3. Téléchargez le fichier XML.

   Utilisez l'une des méthodes décrites dans la section «Téléchargement de fichiers XML», à la page 412.

   Pour télécharger le fichier à partir de la ligne de commande avec l'utilitaire cqupload.jar, procédez comme suit :

   ```
   java -jar /CognosNow/CQUpload/cqupload.jar
   ```

Exemple : Création d'un utilisateur

Cet exemple présente une définition XML destinée à créer un utilisateur.

```
<?xml version="1.0" encoding="UTF-8"?>
<createUser
   xsi:schemaLocation="http://cognos.obi.com/5
                      ../../api/metadata/createUser.xsd"
   xmlns="http://cognos.obi.com/2 "
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
   <name>Skyler</name>
   <description xml:space="preserve">A power user</description>
   <Password>roo</Password>
```

```
</createUser>
```

Définition de plusieurs objets avec le langage XML

Lors de la définition de plusieurs objets, notamment ceux pourvus de dépendances, utilisez l'objet de définition XML en mode de traitement par lots commandBatch.xsd.

Si vous utilisez le mode lot, insérez tout le langage XML dans un seul fichier, puis téléchargez ce fichier. Toutes les opérations doivent être valides, faute de quoi aucune d'elles ne sera acceptée. Pour définir et télécharger un seul fichier XML, suivez les instructions de la section «Définition d'un objet avec le langage XML», à la page 414.

Procédure

1. Créez un fichier de commandes XML. Utilisez la définition commandBatch.xsd. Dans le fichier, emboîtez chaque définition dans un élément <commande>, puis placez-les toutes dans un seul élément <commandes> dans l'ordre dans lequel les objets doivent être définis.

   Pour plus d'informations sur la liste qui définit plusieurs objets dépendants, voir «Exemple : Commande par lots».
2. Assurez-vous de l'exécution de Cognos Real-time Monitoring.
3. Téléchargez le fichier de commandes XML.

   A l'aide de l'utilitaire cqupload.jar, téléchargez le fichier XML sur le serveur d'applications exécutant Real-time Monitoring. Par exemple, téléchargez l'exemple de fichier de commandes :

   ```
   java -jar /CognosNow/CQUpload/cqupload.jar
   -s http://<applicationServer>/now commandBatchSkyler.xml
   ```

Exemple : Commande par lots

Cette commande par lots définit un compte utilisateur, deux profils de diffusion pour l'utilisateur ainsi qu'une préférence utilisateur, puis attribue une autorisation à l'utilisateur.

```
<?xml version="1.0" encoding="UTF-8"?>
<commandBatch
   xsi:schemaLocation="http://cognos.obi.com/5
                       ../../api/metadata/commandBatch.xsd"
    xmlns="http://cognos.obi.com/5"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <commands>
    <command>
      <createUser>
        <name>Skyler</name>
        <description xml:space="preserve">A power
user</description>
        <Password>roo</Password>
      </createUser>
    </command>
    <command>
      <createUserProfile>
```

```
        <name xml:space="preserve">Work e-mail</name>
        <UserName>Skyler</UserName>
        <isDefault>true</isDefault>
        <EmailProfile><typeName/>
          <emailAddress>skyler@email.com</emailAddress>
        </EmailProfile>
      </createUserProfile>
    </command>
    <command>
      <createUserProfile>
        <name xml:space="preserve">Second profile</name>
        <UserName>Skyler</UserName>
        <isDefault>false</isDefault>
        <EmailProfile><typeName/>
          <emailAddress>skyler@viewceler.com</emailAddress>
        </EmailProfile>
      </createUserProfile>
    </command>
    <command>
      <setUserPreferences>
        <userName>Skyler</userName>
        <userPreference>
          <operation>set</operation>
          <name>polling_interval</name>
          <value>5</value>
        </userPreference>
      </setUserPreferences>
    </command>
    <command>
      <setPrivilege>
        <operation>GRANT</operation>
        <privilegeTuple>
          <objectType>VIEW</objectType>
          <privilege>UPDATE</privilege>
          <accessorType>USER</accessorType>
          <accessorName>Skyler</accessorName>
        </privilegeTuple>
        <withGrant>false</withGrant>
      </setPrivilege>
    </command>
  </commands>
</commandBatch>
```

Modification d'un objet existant avec le langage XML

Pour modifier la définition d'un objet existant, utilisez le même XML et le même schéma que ceux utilisés lors de sa création, mais insérez un élément <alterInformation> afin d'identifier la modification.

(L'élément <alterInformation> est défini dans le fichier common.xsd.) Par exemple, cette définition renomme une vue OldName en NewName, à l'aide du schéma createView.xsd :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<createView xsi:schemaLocation="http://cognos.obi.com/5
    /api/metadata/createView.xsd"
    xmlns="http://cognos.obi.com/5"     xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
>
    <alterInformation>
        <previousName> OldName
</previousName>
    </alterInformation>
    <name>
newName
</name>
    <description>My test view</description>
    <query>SELECT c1, c2 FROM event1 WHERE c3&lt;=100</query>
</createView>
```

Quelle que soit la modification que vous implémentez, vous devez utiliser <previousName>. Si vous ne changez pas le nom de l'objet, utilisez un même nom pour les deux éléments <previousName> et <name>, comme indiqué dans l'exemple suivant :

```
    ...
    <alterInformation>
        <previousName>OldName</previousName>
    </alterInformation>
    <name>OldName</name>
    <description> New description </description>
    <query>SELECT c1, c2 FROM event1 WHERE c3&lt;
555 </query>
    ...
```

Par défaut, une modification échoue si l'objet existant n'existe pas. Toutefois, vous pouvez forcer sa création, quelle que soit son existence, en insérant un élément <createIfNotFound>, comme indiqué dans l'exemple suivant :

```
    ...
    <alterInformation>
        <previousName>OldName</previousName>
        <createIfNotFound>true</createIfNotFound>
    </alterInformation>
    ...
```

Dépendances

Si vous modifiez un objet existant, tous les autres objets qui en dépendent sont évalués et rendus incorrects si leur définition est interrompue, conséquence de la modification.

Vous devez modifier les objets incorrects et corriger leur définition pour pouvoir les réactiver.

Emission de commandes XML

De nombreuses opérations système peuvent être exécutées à l'aide de commandes XML.

Le tableau ci-dessous contient certaines des opérations courantes.

Tableau 111. Opérations XML courantes

Opération	Schéma
Activation d'un objet	enableObject.xsd
Désactivation d'un objet	disableObject.xsd
Suppression d'un objet	dropObject.xsd
Définition d'une propriété système	setProperty.xsd
Importation ou exportation des métadonnées système	performImportExport.xsd
Exécution d'un point de contrôle	systemCommand.xsd

Vous trouverez ci-dessous un exemple d'opération.

Exemple : Activation d'un objet et de ses dépendances

Pour activer un objet et toutes ses dépendances, utilisez le schéma enableObject.xsd.

Vous devez identifier le nom de l'objet ainsi que son type. Les valeurs <type> valides sont définies dans le fichier common.xsd par l'élément <VCEnableObjectType>.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enableObject xsi:schemaLocation="http://cognos.obi.com/5
    /api/metadata/enableObject.xsd"
    xmlns="http://cognos.obi.com/5"     xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
>
    <name>MyView</name>
    <type>VIEW</type>
    <cascade>true</cascade>
</enableObject>
```

Cet exemple active tous les objets dépendants, car l'attribut <cascade> est défini sur true. Omettez cet élément ou définissez-le sur false pour activer uniquement l'objet nommé.

Fichiers XSD Cognos Real-time Monitoring

Les fichiers schéma XML IBM Cognos Real-time Monitoring peuvent être classés par catégories.

Ces catégories sont les suivantes :

- Utilisateurs, profils et rôles
- Tableau de bord
- Règles, alertes et tableaux récapitulatifs

- Flux de données, tables de correspondance, cubes et vues
- Scénarios et activités métier
- Agents
- Administration du système
- Gestion des objets
- Fichiers divers

Les fichiers se trouvent à l'emplacement suivant : `realtime\sdk\api\metadata`.

Utilisateurs, profils et rôles

Le tableau suivant comprend la description des fichiers schéma des utilisateurs, des profils et des rôles.

Tableau 112. Fichiers schéma pour les utilisateurs, profils et rôles

Schéma	Description
addMembersToRole.xsd	Ajoute un ou plusieurs utilisateurs existants à un rôle existant.
addMemberToRole.xsd	Ajoute un utilisateur existant à un rôle existant.
createRole.xsd	Crée un objet de rôle d'utilisateur.
createSecurityFilter.xsd	Crée un filtre d'accès.
createUser.xsd	Crée un objet utilisateur.
createUserProfile.xsd	Crée un objet profil de diffusion utilisateur.
setPrivilege.xsd	Définit une autorisation d'utilisateur ou de rôle sur un objet ou une classe d'objets.
setUserPreferences.xsd	Définit les préférences d'un utilisateur.

Tableau de bord

Le tableau suivant comprend la description des fichiers schéma des tableaux de bord.

Tableau 113. Fichiers schéma pour les tableaux de bord

Schéma	Description
createBookmarks.xsd	Crée un signet pour un tableau de bord.
createDashboard.xsd	Crée un tableau de bord.
createPlan.xsd	Crée des références et des seuils de tableaux de bord.
createPortlet.xsd	Crée un objet de tableau de bord.
createTask.xsd	Crée une tâche de tableau de bord.
createTaskMessage.xsd	Crée un message de tâche de tableau de bord.

Règles, alertes et tableaux récapitulatifs

Le tableau suivant comprend la description des fichiers schéma des règles, des alertes et des tableaux récapitulatifs.

Tableau 114. Fichiers schéma pour les règles, alertes et tableaux récapitulatifs

Schéma	Description
alertCommon.xsd	Définitions XSD d'alertes courantes.

Tableau 114. Fichiers schéma pour les règles, alertes et tableaux récapitulatifs (suite)

Schéma	Description
alertMessage.xsd	Message d'alerte à envoyer à un service Web.
alterAlertState.xsd	Modifie l'état d'une alerte existante.
alterRule.xsd	Modifie un objet de règle existant.
createAlert.xsd	Crée un objet d'alerte.
createReportlet.xsd	Crée un objet de tableau récapitulatif.
createRule.xsd	Crée un objet de règle.
createRuleBundle.xsd	Spécifie les valeurs associées aux paramètres d'un modèle de règle et génère les règles, les alertes et les tableaux récapitulatifs en fonction de la définition du modèle.
createRuleTemplate.xsd	Crée un modèle de définitions paramétrées d'un ensemble de règles, l'alerte utilisée par les règles ainsi que les tableaux récapitulatifs associés à l'alerte.
ruleCommon.xsd	Contient les définitions XSD de règles courantes.

Flux de données, tables de correspondance, cubes et vues

Le tableau suivant comprend la description des fichiers schéma des flux de données, des tables de correspondance, des cubes et des vues.

Tableau 115. Fichiers de schéma pour les flux de données, tables de correspondance, cubes et vues

Schéma	Description
createConsolidatedEvent View.xsd	Crée un flux de données consolidées basé sur un flux existant et un ou plusieurs flux de données ou vues supplémentaires.
createContext.xsd	Crée un objet de table de correspondance basé sur un source existante.
createCube.xsd	Crée un objet de cube.
createDimension.xsd	Crée une dimension de cube.
createEventStream.xsd	Crée un objet de flux de données basé sur une source existante.
createView.xsd	Crée un objet de vue basé sur une table source existante (flux de données) et d'autres tables jointes facultatives (table de correspondance ou de flux de données).
createViewPersistence.xsd	Crée une définition de sauvegarde de vue.
ffsourceType.xsd	Objet de flux de données de fichier à plat.
jdbcSource.xsd	Objet de table de correspondance JDBC.
messageSource.xsd	Objet message transmis à partir d'un agent vers une table de correspondance ou de flux de données, utilisé par JMS, TIBCO RV, log4j et le protocole HTTP.
queryCube.xsd	Exécute une requête sur un cube au sujet des mesures d'une dimension.
queryInformation.xsd	Contient les éléments d'une requête (instruction SELECT).
sourceDefinition.xsd	Types source (types d'agent) pris en charge par le système.
sourceDefinition.xsd	Contient les définitions source communes associées aux définitions XSD d'objet de table de correspondance ou de flux de données.

Tableau 115. Fichiers de schéma pour les flux de données, tables de correspondance, cubes et vues (suite)

Schéma	Description
tableDefinition.xsd	Prend en charge la définition d'une source d'événement.
webServiceSource.xsd	Objet de flux de données de service Web.

Scénarios et activités métier

Le tableau suivant comprend la description des fichiers schéma des scénarios et des activités métier.

Tableau 116. Fichiers schéma pour les scénarios et activités métier

Schéma	Description
createBusinessActivity.xsd	Crée un objet d'activité métier.
createScenario.xsd	Crée un objet de scénario dans une vue métier existante et liée à une vue existante.

Agents

Le tableau suivant comprend la description des fichiers schéma des agents.

Tableau 117. Fichiers schéma pour les agents

Schéma	Description
FFConnection.xsd	Agent de fichier à plat.
jdbcConnection.xsd	Agent JDBC.
JMSTopicConnection.xsd	Agent JMS.
log4jConnection.xsd	Agent de messagerie log4j (utilisé en interne pour la journalisation).
procSource.xsd	Définitions des procédures stockées.
RVConnection.xsd	Agent TIBCO RV.
SOAPConnection.xsd	Connexion à un service Web.
xmlBuffer.xsd	Mémoire tampon XML qui contient une partie du message.

Administration du système

Le tableau suivant comprend la description des fichiers schéma d'administration du système.

Tableau 118. Fichiers schéma pour l'administration de système

Schéma	Description
propertyTypeDefinition.xsd	Décrit une propriété système.
setLogLevel.xsd	Définit le niveau de journalisation d'un consignateur de système (module de journalisation).
setProperty.xsd	Commande permettant de définir une propriété système.
systemCommand.xsd	Exécute un point de contrôle.

Gestion des objets

Le tableau suivant comprend la description des fichiers schéma de gestion des objets.

Tableau 119. Fichiers schéma pour la gestion des objets

Schéma	Description
createKeyRelationship.xsd	Crée une relation entre deux objets.
disableObject.xsd	Désactive un objet.
dropObject.xsd	Supprime un objet.
enableObject.xsd	Active un objet.
setObjectRelation.xsd	Relie deux objets.

Fichiers divers

Le tableau suivant comprend la description de différents fichiers schéma.

Tableau 120. Fichiers schéma divers

Schéma	Description
commandBatch.xsd	Définit plusieurs objets à charger (définir) dans un lot. Toutes les définitions doivent être valides, faute de quoi aucun objet ne sera défini.
common.xsd	Contient les définitions XSD courantes utilisées par la plupart des schémas XSD.
createJar.xsd	Crée un objet JAR.
createUDF.xsd	Crée un objet UDF.
invokeExternalAction.xsd	Décrit un message d'action externe envoyé à un service Web. Pour plus d'informations sur ce fichier, voir «Processus externes des services Web», à la page 386.
jarManifest.xsd	Définit le fichier manifeste dans un fichier JAR.
performImportExport.xsd	Indique au système d'exécuter une importation ou une exportation.
schedule.xsd	Contient les définitions des calendriers courantes et les définitions d'intervalle de programmation utilisées par les définitions XSD.

Annexe A. Méthodes JDBC (Java Database Connectivity)

La présente annexe répertorie les classes et les méthodes du pilote IBM Cognos Real-time Monitoring destiné à être utilisé avec JDBC. La description associée à chaque méthode indique si elle est conforme à la norme API JDBC ou renvoie des données qui sont différentes de la norme JDBC 2.0.

Driver

Pilote de la base de données SQL JDBC.

La classe est nommée `com.cognos.obi.jdbc.driver.Driver`

`boolean acceptsURL(String url)`

Renvoie la valeur True si l'URL commence par "jdbc:cognos:obi:".

`Connection connect(String url, Properties info)`

`int getMajorVersion()`

`int getMinorVersion()`

`DriverPropertyInfo[] getPropertyInfo(Stringurl,Properties info)`

Renvoie `null`.

`boolean jdbcCompliant()`

Renvoie `false`.

Connection

Connexion à une base de données particulière.

`void clearWarnings()`

Aucune action.

`void close()`

Compatible avec l'API JDBC standard.

`void commit()`

Aucune action. Le pilote JDBC IBM Cognos Real-time Monitoring est en lecture seule. Le concept de validation n'est pas applicable.

`Statement createStatement()`

Compatible avec l'API JDBC standard.

© Copyright IBM Corp. 2007, 2014

StatementcreateStatement(int resultSetType,int resultSetConcurrency)

Compatible avec l'API JDBC standard lorsque `resultSetType` est `ResultSet.TYPE_FORWARD_ONLY` et `resultSetConcurrency` est `ResultSet.CONCUR_READ_ONLY`. Sinon, une exception `SQLException` est lancée, indiquant que la méthode n'est pas implémentée.

Statement createStatement(int resultSetType,int resultSetConcurrency, int resultSetHoldability)

Non implémenté. Génère une exception `SQLException`.

boolean getAutoCommit()

Renvoie `false`. Le pilote JDBC IBM Cognos Real-time Monitoring est en lecture seule. Le concept de validation n'est pas applicable.

String getCatalog()

Renvoie "IBM Cognos Real-time Monitoring"

int getHoldability()

Renvoie `ResultSet.HOLD_CURSORS_OVER_COMMIT`.

Avertissement : La méthode `commit()` n'est pas opérationnelle dans le pilote Real-time Monitoring. Par conséquent, un ensemble `ResultSet` n'est pas fermé lorsque la méthode `Connection.commit()` est appelée.

DatabaseMetaData getMetaData()

Compatible avec l'API JDBC standard. L'objet `DatabaseMetadata` renvoyé par cette méthode est également une instance de l'objet `IOlapMetadataData`

int getTransactionIsolation()

Renvoie `TRANSACTION_NONE`.

Map getTypeMap()

Non implémenté. Génère une exception `SQLException`.

SQLWarning getWarnings()

Renvoie `null`.

Connexion

Connexion à une base de données particulière.

void clearWarnings()

Aucune action.

void close()

Compatible avec l'API JDBC standard.

void commit()

Aucune action. Le pilote JDBC IBM Cognos Real-time Monitoring est en lecture seule. Le concept de validation n'est pas applicable.

Statement createStatement()

Compatible avec l'API JDBC standard.

StatementcreateStatement(int resultSetType, int resultSetConcurrency)

Compatible avec l'API JDBC standard lorsque `resultSetType` est `ResultSet.TYPE_FORWARD_ONLY` et que `resultSetConcurrency` est `ResultSet.CONCUR_READ_ONLY`. Sinon, une exception `SQLException` est lancée, indiquant que la méthode n'a pas été implémentée.

Statement createStatement(int resultSetType, int resultSetConcurrency, int resultSetHoldability)

Non implémenté. Génère une exception `SQLException`.

boolean getAutoCommit()

Renvoie `false`. Le pilote JDBC IBM Cognos Real-time Monitoring est en lecture seule. Le concept de validation n'est pas applicable.

String getCatalog()

Renvoie "IBM Cognos Real-time Monitoring"

int getHoldability()

Renvoie `ResultSet.HOLD_CURSORS_OVER_COMMIT`.

Avertissement : La méthode `commit()` n'est pas opérationnelle dans le pilote Cognos Real-time Monitoring. Par conséquent, un résultat `ResultSet` n'est pas fermé lorsque la méthode `Connection.commit()` est appelée.

DatabaseMetaData getMetaData()

Compatible avec l'API JDBC standard. L'objet `DatabaseMetadata` renvoyé par cette méthode est également une instance de l'objet `IOlapMetadataData`

int getTransactionIsolation()

Renvoie `TRANSACTION_NONE`.

Map getTypeMap()

Non implémenté. Génère une exception `SQLException`.

SQLWarning getWarnings()

Renvoie `null`.

boolean isClosed()

Compatible avec l'API JDBC standard.

boolean isReadOnly()

Renvoie `true`.

String nativeSQL(String sql)

Non implémenté. Génère une exception `SQLException`.

CallableStatement prepareCall(String sql)

Non implémenté. Génère une exception `SQLException`.

CallableStatement prepareCall(String sql, int resultSetType, int resultSetConcurrency)

Non implémenté. Génère une exception `SQLException`.

PreparedStatement prepareStatement(String sql)

Compatible avec l'API JDBC standard.

PreparedStatement prepareStatement(String sql, int[] columnIndexes)

L'argument `autoGeneratedKeys` est ignoré.

PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)

Les arguments `resultSetType` et `resultSetConcurrency` sont ignorés.

PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency, int resultSetHoldability)

Les arguments `resultSetType`, `resultSetConcurrency` et `resultSetHoldability` sont ignorés.

PreparedStatement prepareStatement(String sql, String[] columnNames)

L'argument `columnNames` est ignoré.

releaseSavepoint(Savepoint savepoint)

Non implémenté. Génère une exception `SQLException`.

void rollback()

Aucune action.

void rollback(Savepoint savepoint)

Non implémenté. Génère une exception `SQLException`.

void setAutoCommit(boolean autoCommit)

Aucune action.

setCatalog(String catalog)

Aucune action.

void setHoldability(int holdability)

Non implémenté. Génère une exception `SQLException`.

void setReadOnly(boolean readOnly)

Non implémenté. Génère une exception `SQLException`.

Savepoint setSavepoint(String name)

Non implémenté. Génère une exception `SQLException`.

Savepoint setSavepoint(String name)

Non implémenté. Génère une exception `SQLException`.

void setTransactionIsolation(int level)

Non implémenté. Génère une exception `SQLException`.

void setTypeMap(Map map)

Non implémenté. Génère une exception `SQLException`.

Statement

Instruction SQL à transmettre à la base de données.

void addBatch(String sql)

Ne pas utiliser. Usage interne uniquement.

void cancel()

Non implémenté. Génère une exception `SQLException`.

void clearBatch()

Ne pas utiliser. Usage interne uniquement.

void clearWarnings()

Aucune action.

void close()

Compatible avec l'API JDBC standard.

boolean execute(String sql)

Compatible avec l'API JDBC standard.

boolean execute(String sql, int autoGeneratedKeys)

Non implémenté. Génère une exception `SQLException`.

boolean execute(String sql, int[] columnIndexes)

Non implémenté. Génère une exception `SQLException`.

boolean execute(String sql, int[] columnNames)

Non implémenté. Génère une exception `SQLException`.

int[] executeBatch()

Ne pas utiliser. Usage interne uniquement.

ResultSet executeQuery(String sql)

Compatible avec l'API JDBC standard.

int executeUpdate(String sql)

Non implémenté. Génère une exception `SQLException`.

int executeUpdate(String sql, int autoGeneratedKeys)

Non implémenté. Génère une exception `SQLException`.

int executeUpdate(String sql, int[] columnIndexes)

Non implémenté. Génère une exception `SQLException`.

int executeUpdate(String sql, int[] columnNames)

Non implémenté. Génère une exception `SQLException`.

Connection getConnection()

Compatible avec l'API JDBC standard.

int getFetchDirection()

Non implémenté. Génère une exception `SQLException`.

int getFetchSize()

Compatible avec l'API JDBC standard.

ResultSet getGeneratedKeys()

Non implémenté. Génère une exception `SQLException`.

getMaxFieldSize()

`Returns 0.`

int getMaxRows()

Compatible avec l'API JDBC standard.

boolean getMoreResults()

Passe au résultat suivant de l'objet `Statement` et renvoie `true` s'il s'agit d'un objet `ResultSet`.

Avertissement : Lorsqu'elle est utilisée avec IBM Cognos Real-time Monitoring, l'API JDBC ne ferme implicitement aucun des objets `ResultSet` en cours obtenus par le biais de la méthode `getResultSet`.

boolean getMoreResults(int current)

Passe au résultat suivant de l'objet `Statement` et renvoie `true` s'il s'agit d'un objet `ResultSet`. La valeur de `current` est ignorée.

Avertissement : Lorsqu'elle est utilisée avec IBM Cognos Real-time Monitoring, l'API JDBC ne ferme implicitement aucun des objets `ResultSet` en cours obtenus par le biais de la méthode `getResultSet`.

int getQueryTimeout()

Non implémenté. Génère une exception `SQLException`.

ResultSet getResultSet()

Compatible avec l'API JDBC standard.

int getResultSetConcurrency()

Non implémenté. Génère une exception `SQLException`.

int getResultSetHoldability()

Non implémenté. Génère une exception `SQLException`.

int getResultSetType()

Non implémenté. Génère une exception `SQLException`.

int getUpdateCount()

Renvoie -1.

SQLWarning getWarnings()

Renvoie `null` si l'instruction `Statement` n'est pas close.

void setCursorName(String name)

Non implémenté. Génère une exception `SQLException`.

void setEscapeProcessing(boolean enable)

Non implémenté. Génère une exception `SQLException`.

void setFetchDirection(int direction)

Non implémenté. Génère une exception `SQLException`.

void setFetchSize(int rows)

Compatible avec l'API JDBC standard.

void setMaxFieldSize(int max)

Non implémenté. Génère une exception `SQLException`.

void setMaxRows(int max)

Compatible avec l'API JDBC standard.

void setQueryTimeout(int seconds)

Non implémenté. Génère une exception `SQLException`.

DatabaseMetaData

Fournit des informations sur les définitions des vues lors de l'installation d' IBM Cognos Real-time Monitoring.

boolean allProceduresAreCallable()

Renvoie `false`.

boolean allTablesAreSelectable()

Renvoie `true`.

boolean dataDefinitionCausesTransactionCommit()

Renvoie `false`.

Remarque : Le pilote JDBC Real-time Monitoring ne prend pas en charge la définition de données.

boolean dataDefinitionIgnoredInTransactions()

Renvoie `true`.

Remarque : Le pilote JDBC IBM Cognos Real-time Monitoring ne prend pas en charge la définition de données.

boolean deletesAreDetected(int type)

Renvoie `false`.

boolean doesMaxRowSizeIncludeBlobs()

Renvoie `false`.

ResultSet getAttributes(String catalog,String schemaPattern, String typeNamePattern, String attributeNamePattern)

Renvoie un objet `ResultSet` vide.

ResultSet getBestRowIdentifier(String catalog, String schema, String table, int scope, boolean nullable)

Renvoie un objet `ResultSet` vide.

ResultSet getCatalogs()

Compatible avec l'API JDBC standard.

String getCatalogSeparator()

Compatible avec l'API JDBC standard.

String getCatalogTerm()

Compatible avec l'API JDBC standard.

ResultSet getColumnPrivileges(String catalog, String schema, String table, String columnNamePattern)

Renvoie un objet `ResultSet` vide.

ResultSet getColumns(String catalog,String schemaPattern, String tableNamePattern, String columnNamePattern)

Extrait la description des colonnes de table disponibles dans la table. L'API JDBC Real-time Monitoring JDBC API ne traite pas les arguments `schemaPattern`, `tableNamePattern` et `columnNamePattern` en tant que modèles. Si la valeur de l'un de ces trois arguments est NULL ou "%", tous les noms correspondent. Sinon, l'argument est traité en tant que nom exact et non en tant que modèle.

Connection getConnection()

Compatible avec l'API JDBC standard.

ResultSet getCrossReference(String primaryCatalog, String primarySchema, String primaryTable, String foreignCatalog, String foreignSchema, String foreignTable)

Renvoie un objet `ResultSet` vide.

int getDatabaseMajorVersion()

Compatible avec l'API JDBC standard.

int getDatabaseMajorVersion()

Compatible avec l'API JDBC standard.

String getDatabaseMinorVersion()

Compatible avec l'API JDBC standard.

String getDatabaseProductName()

Compatible avec l'API JDBC standard.

String getDatabaseProductVersion()

Compatible avec l'API JDBC standard.

int getDefaultTransactionIsolation()

Renvoie `Connection.TRANSACTION_NONE`.

int getDriverMajorVersion()

Compatible avec l'API JDBC standard.

int getDriverMinorVersion()

Compatible avec l'API JDBC standard.

String getDriverName()

Compatible avec l'API JDBC standard.

String getDriverVersion()

Compatible avec l'API JDBC standard.

ResultSet getExportedKeys(String catalog, String schema, String table)

Renvoie un objet `ResultSet` vide.

String getExtraNameCharacters()

Renvoie une chaîne vide.

String getIdentifierQuoteString()

Compatible avec l'API JDBC standard.

ResultSet getImportedKeys(String catalog, String schema, String table)

Renvoie un objet `ResultSet` vide.

ResultSet getIndexInfo(String catalog, String schema, String table, boolean unique, boolean approximate)

Renvoie un objet `ResultSet` vide.

int getJDBCMajorVersion()

Compatible avec l'API JDBC standard.

int getJDBCMinorVersion()

Compatible avec l'API JDBC standard.

int getMaxBinaryLiteralLength()

Renvoie 0.

int getMaxCatalogNameLength()

Renvoie 32.

int getMaxCharLiteralLength()

Renvoie 0.

int getMaxColumnNameLength()

Renvoie 128.

int getMaxColumnsInGroupBy()

Renvoie 0.

int getMaxColumnsInIndex()

Renvoie 0.

int getMaxColumnsInOrderBy()

Renvoie 0.

int getMaxColumnsInSelect()

Renvoie 0.

int getMaxColumnsInTable()

Renvoie 0.

int getMaxConnections()

Renvoie 0.

int getMaxCursorNameLength()

Renvoie 0.

int getMaxIndexLength()

Renvoie 0.

int getMaxProcedureNameLength()

Renvoie 0.

int getMaxRowSize()

Renvoie 0.

int getMaxSchemaNameLength()

Renvoie 0.

int getMaxStatementLength()

Renvoie 0.

int getMaxStatements()

Renvoie 0.

int getMaxTableNameLength()

Renvoie 128.

int getMaxTablesInSelect()

Renvoie 0.

int getMaxUserNameLength()

Renvoie 128.

String getNumericFunctions()

Renvoie "avg, sum, count, max, min".

ResultSet getPrimaryKeys(String catalog, String schema, String table)

Renvoie un objet `ResultSet` vide.

ResultSet getProcedureColumns(String catalog, String schemaPattern, String procedureNamePattern, String columnNamePattern)

Renvoie un objet `ResultSet` vide.

ResultSet getProcedures(String catalog, String schemaPattern, String procedureNamePattern)

Renvoie un objet `ResultSet` vide.

String getProcedureTerm()

Renvoie "Procedure".

int getResultSetHoldability()

Renvoie `ResultSet.HOLD_CURSORS_OVER_COMMIT`.

ResultSet getSchemas()

Compatible avec l'API JDBC standard.

String getSchemaTerm()

Renvoie "schema".

String getSearchStringEscape()

Renvoie `null`.

String getSQLKeywords()

Renvoie une `String` vide.

int getSQLStateType()

Renvoie 0.

String getStringFunctions()

Renvoie une `String` vide.

ResultSet getSuperTables(String catalog, String schemaPattern, String tableNamePattern)

Renvoie un objet `ResultSet` vide.

ResultSet getSuperTypes(String catalog, String schemaPattern, String typeNamePattern)

Renvoie un objet `ResultSet` vide.

String getSystemFunctions()

Renvoie une `String` vide.

ResultSet getTablePrivileges(String catalog, String schemaPattern, String tableNamePattern)

Renvoie un objet `ResultSet` vide.

ResultSet getTables(String catalog,String schemaPattern, String tableNamePattern, String[] types)

Les arguments `schemaPattern` et `tableNamePattern` ne sont pas traités en tant que modèles. Si la valeur de l'un ou l'autre de ces arguments est `null` ou "%", tous les noms correspondent. Sinon, l'argument est traité en tant que nom exact et non en tant que modèle.

ResultSet getTableTypes()

Compatible avec l'API JDBC standard.

String getTimeDateFunctions()

Renvoie une `String` vide.

ResultSet getTypeInfo()

Renvoie un objet `ResultSet` vide.

ResultSet getUDTs(String catalog, String schemaPattern, String typeNamePattern, int[] types)

Renvoie un objet `ResultSet` vide.

String getURL()

Renvoie `null`.

String getUserName()

Compatible avec l'API JDBC standard.

ResultSet getVersionColumns(String catalog, String schema, String table)

Renvoie un objet `ResultSet` vide.

boolean insertsAreDetected(int type)

Compatible avec l'API JDBC standard.

boolean isCatalogAtStart()

Compatible avec l'API JDBC standard.

boolean isReadOnly()

Compatible avec l'API JDBC standard.

boolean locatorsUpdateCopy()

Renvoie true.

boolean nullPlusNonNullIsNull()

Compatible avec l'API JDBC standard.

boolean nullsAreSortedAtEnd()

Compatible avec l'API JDBC standard.

boolean nullsAreSortedAtStart()

Compatible avec l'API JDBC standard.

boolean nullsAreSortedHigh()

Compatible avec l'API JDBC standard.

boolean nullsAreSortedLow()

Compatible avec l'API JDBC standard.

boolean othersDeletesAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean othersInsertsAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean othersUpdatesAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean ownDeletesAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean ownInsertsAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean ownUpdatesAreVisible(int type)

Compatible avec l'API JDBC standard.

boolean storesLowerCaseIdentifiers()

boolean storesLowerCaseQuotedIdentifiers()

Compatible avec l'API JDBC standard.

boolean storesMixedCaseIdentifiers()

Compatible avec l'API JDBC standard.

boolean storesMixedCaseQuotedIdentifiers()

Compatible avec l'API JDBC standard.

boolean storesUpperCaseIdentifiers()

Compatible avec l'API JDBC standard.

boolean storesUpperCaseQuotedIdentifiers()

Compatible avec l'API JDBC standard.

boolean supportsAlterTableWithAddColumn()

Renvoie `false`.

boolean supportsAlterTableWithDropColumn()

Renvoie `false`.

boolean supportsANSI92EntryLevelSQL()

Renvoie `false`.

boolean supportsANSI92FullSQL()

Renvoie `false`.

boolean supportsANSI92IntermediateSQL()

Renvoie `false`.

boolean supportsBatchUpdates()

Renvoie `false`.

boolean supportsCatalogsInDataManipulation()

Renvoie `true`.

boolean supportsCatalogsInIndexDefinitions()

Renvoie `false`.

boolean supportsCatalogsInPrivilegeDefinitions()

Renvoie `false`.

boolean supportsCatalogsInProcedureCalls()

Renvoie `false`.

boolean supportsCatalogsInTableDefinitions()

Renvoie true.

boolean supportsColumnAliasing()

Extrait si cette base de données prend en charge les alias de colonne.

boolean supportsConvert(int fromType, int toType)

Renvoie false.

supportsConvert(int fromType, int toType)

Renvoie false.

boolean supportsCoreSQLGrammar()

Renvoie false.

boolean supportsCorrelatedSubqueries()

Renvoie false.

boolean supportsDataDefinitionAndDataManipulationTransactions()

Renvoie false.

boolean supportsDataManipulationTransactionsOnly()

Renvoie false.

boolean supportsDifferentTableCorrelationNames()

Renvoie false.

boolean supportsExpressionsInOrderBy()

Renvoie false.

boolean supportsExtendedSQLGrammar()

Renvoie false.

boolean supportsFullOuterJoins()

Renvoie false.

boolean supportsGetGeneratedKeys()

Renvoie false.

boolean supportsGroupBy()

Renvoie true.

boolean supportsGroupByBeyondSelect()

Renvoie `true`.

boolean supportsGroupByUnrelated()

Renvoie `true`.

boolean supportsIntegrityEnhancementFacility()

Renvoie `false`.

boolean supportsLikeEscapeClause()

Renvoie `true`.

boolean supportsLimitedOuterJoins()

Renvoie `true`.

boolean supportsMinimumSQLGrammar()

Renvoie `false`.

boolean supportsMixedCaseIdentifiers()

Renvoie `false`.

boolean supportsMixedCaseQuotedIdentifiers()

Renvoie `false`.

boolean supportsMultipleOpenResults()

Renvoie `false`.

boolean supportsMultipleResultSets()

Renvoie `false`.

boolean supportsMultipleTransactions()

Renvoie `false`.

boolean supportsNamedParameters()

Renvoie `false`.

boolean supportsNonNullableColumns()

Renvoie `false`.

boolean supportsOpenStatementsAcrossCommit()

Renvoie `true`.

boolean supportsOpenStatementsAcrossRollback()

Renvoie true.

boolean supportsOpenStatementsAcrossCommit()

Renvoie true.

boolean supportsOpenStatementsAcrossRollback()

Renvoie false.

boolean supportsOrderByUnrelated()

Renvoie false.

boolean supportsOuterJoins()

Renvoie true.

boolean supportsPositionedDelete()

Renvoie false.

boolean supportsPositionedUpdate()

Renvoie false.

boolean supportsResultSetConcurrency(int type, int concurrency)

Renvoie false.

boolean supportsResultSetHoldability(int holdability)

Renvoie true si holdability est ResultSet.HOLD_CURSORS_OVER_COMMIT.

boolean supportsResultSetType(int type)

Renvoie true si type est ResultSet.TYPE_FORWARD_ONLY.

boolean supportsSavepoints()

Renvoie false.

boolean supportsSchemasInDataManipulation()

Renvoie false.

boolean supportsSchemasInIndexDefinitions()

Renvoie false.

boolean supportsSchemasInPrivilegeDefinitions()

Renvoie false.

boolean supportsSchemasInProcedureCalls()

Renvoie `false`.

boolean supportsSchemasInTableDefinitions()

Renvoie `false`.

boolean supportsSelectForUpdate()

Renvoie `false`.

boolean supportsStatementPooling()

Renvoie `false`.

boolean supportsStoredProcedures()

Renvoie `false`.

boolean supportsSubqueriesInComparisons()

Renvoie `false`.

boolean supportsSubqueriesInExists()

Renvoie `false`.

boolean supportsSubqueriesInIns()

Renvoie `false`.

boolean supportsSubqueriesInQuantifieds()

Renvoie `false`.

boolean supportsTableCorrelationNames()

Renvoie `true`.

boolean supportsTransactionIsolationLevel(int level)

Renvoie `true` si `level` est `Connection.TRANSACTION_NONE`.

boolean supportsTransactions()

Renvoie `false`.

boolean supportsUnion()

Renvoie `false`.

boolean supportsUnionAll()

Renvoie `false`.

boolean updatesAreDetected(int type)

Renvoie `false`.

boolean usesLocalFilePerTable()

Renvoie `false`.

boolean usesLocalFiles()

Renvoie `false`.

ResultSet

Table de données représentant un résultat de la base de données qui est généré lors de l'exécution d'une instruction interrogeant la base de données.

boolean absolute(int row)

Non implémenté. Génère une exception `SQLException`.

void afterLast()

Non implémenté. Génère une exception `SQLException`.

void beforeFirst()

Non implémenté. Génère une exception `SQLException`.

void cancelRowUpdates()

Non implémenté. Génère une exception `SQLException`.

void clearWarnings()

Non implémenté. Génère une exception `SQLException`.

void close()

Compatible avec l'API JDBC standard.

void deleteRow()

Non implémenté. Génère une exception `SQLException`.

int findColumn(String columnName)

Compatible avec l'API JDBC standard.

boolean first()

Non implémenté. Génère une exception `SQLException`.

Array getArray(int i)

Non implémenté. Génère une exception `SQLException`.

Array getArray(String colName)

Non implémenté. Génère une exception `SQLException`.

InputStream getAsciiStream(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

InputStream getAsciiStream(int columnNAME)

Non implémenté. Génère une exception `SQLException`.

BigDecimal getBigDecimal(int columnIndex)

Compatible avec l'API JDBC standard.

BigDecimal getBigDecimal(int columnIndex, int scale)

Non implémenté. Génère une exception `SQLException`.

getBigDecimal(String columnName)

Compatible avec l'API JDBC standard.

InputStream getBinaryStream(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

InputStream getBinaryStream(String columnName)

Non implémenté. Génère une exception `SQLException`.

Blob getBlob(int i)

Non implémenté. Génère une exception `SQLException`.

Blob getBlob(String colName)

Non implémenté. Génère une exception `SQLException`.

boolean getBoolean(String columnIndex)

Compatible avec l'API JDBC standard.

boolean getBoolean(String columnName)

Compatible avec l'API JDBC standard.

byte getByte(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

byte getByte(String columnName)

Non implémenté. Génère une exception `SQLException`.

byte[] getBytes(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

byte[] getBytes(String columnName)

Non implémenté. Génère une exception `SQLException`.

Reader getCharacterStream(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

Reader getCharacterStream(int columnName)

Non implémenté. Génère une exception `SQLException`.

Clob getClob(int i)

Non implémenté. Génère une exception `SQLException`.

Clob getClob(String colName)

Non implémenté. Génère une exception `SQLException`.

int getConcurrency()

Non implémenté. Génère une exception `SQLException`.

String getCursorName()

Non implémenté. Génère une exception `SQLException`.

Date getDate(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

Date getDate(int columnName)

Date getDate(int columnIndex, Calendar cal)

Non implémenté. Génère une exception `SQLException`.

double getDouble(int columnIndex)

Compatible avec l'API JDBC standard.

double getDouble(String columnName)

Compatible avec l'API JDBC standard.

int getFetchDirection()

Non implémenté. Génère une exception `SQLException`.

int getFetchSize()

Compatible avec l'API JDBC standard.

float getFloat(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

float getFloat(String columnName)

Non implémenté. Génère une exception `SQLException`.

int getInt(int columnIndex)

Compatible avec l'API JDBC standard.

int getInt(String columnName)

Compatible avec l'API JDBC standard.

long getLong(int columnIndex)

Compatible avec l'API JDBC standard.

ResultSetMetaData getMetaData()

Compatible avec l'API JDBC standard.

Object getObject(int columnIndex)

Compatible avec l'API JDBC standard.

Object getObject(int i, Map map)

Non implémenté. Génère une exception `SQLException`.

Object getObject(String columnName)

Compatible avec l'API JDBC standard.

Object getObject(String colName, Map map)

Non implémenté. Génère une exception `SQLException`.

Ref getRef(int i)

Non implémenté. Génère une exception `SQLException`.

int getRow()

Non implémenté. Génère une exception `SQLException`.

short getShort(int columnIndex)

Compatible avec l'API JDBC standard.

short getStatement()

Compatible avec l'API JDBC standard.

String getString(int columnIndex)

Compatible avec l'API JDBC standard.

String getString(String columnName)

Compatible avec l'API JDBC standard.

Time getTime(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

Time getTime(int columnName)

Non implémenté. Génère une exception `SQLException`.

Time getTime(int columnIndex, Calendar cal)

Non implémenté. Génère une exception `SQLException`.

Timestamp getTimestamp(int columnIndex)

Compatible avec l'API JDBC standard.

Timestamp getTimestamp(int columnIndex, Calendar cal)

Non implémenté. Génère une exception `SQLException`.

Timestamp getTimestamp(String columnName)

Compatible avec l'API JDBC standard.

Timestamp getTimestamp(String columnName, Calendar cal)

Non implémenté. Génère une exception `SQLException`.

int getType()

Non implémenté. Génère une exception `SQLException`.

InputStream getUnicodeStream(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

InputStream getUnicodeStream(int columnName)

Non implémenté. Génère une exception `SQLException`.

URL getURL(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

URL getURL(String columnName)

Non implémenté. Génère une exception `SQLException`.

SQLWarning getWarnings()

Non implémenté. Génère une exception `SQLException`.

void insertRow()

Non implémenté. Génère une exception `SQLException`.

boolean isAfterLast()

Non implémenté. Génère une exception `SQLException`.

boolean isBeforeFirst()

Non implémenté. Génère une exception `SQLException`.

boolean isFirst()

Non implémenté. Génère une exception `SQLException`.

boolean isLast()

Non implémenté. Génère une exception `SQLException`.

boolean last()

Non implémenté. Génère une exception `SQLException`.

void moveToCurrentRow()

Non implémenté. Génère une exception `SQLException`.

void moveToInsertRow()

Non implémenté. Génère une exception `SQLException`.

boolean next()

Compatible avec l'API JDBC standard.

boolean previous()

Non implémenté. Génère une exception `SQLException`.

void refreshRow()

Non implémenté. Génère une exception `SQLException`.

boolean relative(int rows)

Non implémenté. Génère une exception `SQLException`.

boolean rowDeleted()

Non implémenté. Génère une exception `SQLException`.

boolean rowInserted()

Non implémenté. Génère une exception `SQLException`.

boolean rowUpdated()

Non implémenté. Génère une exception `SQLException`.

void setFetchDirection(int direction)

Non implémenté. Génère une exception `SQLException`.

void setFetchSize(int rows)

Compatible avec l'API JDBC standard.

void updateArray(int columnIndex, Array x)

Non implémenté. Génère une exception `SQLException`.

void updateArray(String columnName, Array x)

Non implémenté. Génère une exception `SQLException`.

void updateAsciiStream(int columnIndex, InputStream x, int length)

Non implémenté. Génère une exception `SQLException`.

void updateAsciiStream(String columnName, InputStream x, int length)

Non implémenté. Génère une exception `SQLException`.

void updateBigDecimal(int columnIndex, BigDecimal x)

Non implémenté. Génère une exception `SQLException`.

void updateBigDecimal(String columnName, BigDecimal x)

Non implémenté. Génère une exception `SQLException`.

void updateBinaryStream(int columnIndex, InputStream x, int length)

Non implémenté. Génère une exception `SQLException`.

void updateBinaryStream(int columnIndex, InputStream x, int length)

Non implémenté. Génère une exception `SQLException`.

void updateBlob(int columnIndex, Blob x)

Non implémenté. Génère une exception `SQLException`.

void updateBlob(String columnName, Blob x)

Non implémenté. Génère une exception `SQLException`.

void updateBoolean(int columnIndex, boolean x)

Non implémenté. Génère une exception `SQLException`.

void updateByte(int columnIndex, byte x)

Non implémenté. Génère une exception `SQLException`.

void updateByte(String columnName, byte x)

Non implémenté. Génère une exception `SQLException`.

void updateBytes(int columnIndex, byte[] x)

Non implémenté. Génère une exception `SQLException`.

void updateBytes(String columnName, byte[] x)

Non implémenté. Génère une exception `SQLException`.

void updateCharacterStream(int columnIndex,Reader x, int length)

Non implémenté. Génère une exception `SQLException`.

void updateCharacterStream(String columnName, Reader reader, int length)

Non implémenté. Génère une exception `SQLException`.

void updateClob(int columnIndex,Clob x)

Non implémenté. Génère une exception `SQLException`.

void updateClob(String columnName, Clob x)

Non implémenté. Génère une exception `SQLException`.

void updateDate(int columnIndex, Date x)

Non implémenté. Génère une exception `SQLException`.

void updateDate(String columnName, Date x)

Non implémenté. Génère une exception `SQLException`.

void updateDouble(int columnIndex, double x)

Non implémenté. Génère une exception `SQLException`.

void updateFloat(int columnIndex, float x)

Non implémenté. Génère une exception `SQLException`.

void updateInt(int columnIndex, int x)

Non implémenté. Génère une exception `SQLException`.

void updateInt(String columnName, int x)

Non implémenté. Génère une exception `SQLException`.

void updateLong(int columnIndex, long x)

Non implémenté. Génère une exception `SQLException`.

void updateLong(String columnName,long x)

Non implémenté. Génère une exception `SQLException`.

void updateNull(int columnIndex)

Non implémenté. Génère une exception `SQLException`.

void updateNull(String columnName)

Non implémenté. Génère une exception `SQLException`.

void updateObject(int columnIndex, Object x)

Non implémenté. Génère une exception `SQLException`.

void updateObject(int columnIndex, Object x, int scale)

Non implémenté. Génère une exception `SQLException`.

void updateObject(String columnName, Object x)

Non implémenté. Génère une exception `SQLException`.

void updateObject(String columnName, Object x, int scale)

Non implémenté. Génère une exception `SQLException`.

void updateRef(int columnIndex, Ref x)

Non implémenté. Génère une exception `SQLException`.

void updateRow()

Non implémenté. Génère une exception `SQLException`.

void updateShort(int columnIndex,short x)

Non implémenté. Génère une exception `SQLException`.

void updateShort(String columnName, short x)

Non implémenté. Génère une exception `SQLException`.

void updateString(int columnIndex, String x)

Non implémenté. Génère une exception `SQLException`.

void updateString(String columnName, String x)

Non implémenté. Génère une exception `SQLException`.

void updateTime(int columnIndex,Time x)

Non implémenté. Génère une exception `SQLException`.

void updateTimestamp(int columnIndex,Timestamp x)

Non implémenté. Génère une exception `SQLException`.

void updateTimestamp(String columnName,Timestamp x)

Non implémenté. Génère une exception `SQLException`.

boolean wasNull()

Non implémenté. Génère une exception `SQLException`.

ResultSetMetaData

Fournit des informations sur les types et les propriétés des colonnes d'un objet ResultSet.

String getCatalogName(int column)

Renvoie "IBM Cognos Real-time Monitoring"

String getColumnClassName(int column)

Renvoie `null`.

int getColumnCount()

Compatible avec l'API JDBC standard.

int getColumnDisplaySize(int column)

Compatible avec l'API JDBC standard.

String getColumnLabel(int column)

Compatible avec l'API JDBC standard.

String getColumnName(int column)

Compatible avec l'API JDBC standard.

int getColumnType(int column)

Compatible avec l'API JDBC standard.

String getColumnTypeName(int column)

Compatible avec l'API JDBC standard.

int getPrecision(int column)

Compatible avec l'API JDBC standard.

int getScale(int column)

Compatible avec l'API JDBC standard.

String getSchemaName(int column)

Renvoie "IBM Cognos Real-time Monitoring".

String getTableName(int column)

Compatible avec l'API JDBC standard.

boolean isAutoIncrement(int column)

Renvoie `false`.

boolean isCaseSensitive(int column)

Renvoie `false`.

boolean isCurrency(int column)

Renvoie `false`.

boolean isDefinitelyWritable(int column)

Renvoie `false`.

boolean isDefinitelyWritable(int column)

Renvoie `false`.

int isNullable(int column)

Renvoie `columnNullable`.

isReadOnly(int column)

Renvoie `true`.

isSearchable(int column)

Compatible avec l'API JDBC standard.

boolean isSigned(int column)

Renvoie `true`.

boolean isWritable(int column)

Renvoie `false`.

Annexe B. API UM-SSO

IBM Cognos Real-time Monitoring comporte une API permettant de créer des plug-ins UM-SSO qui fournissent des fonctions de gestion de l'utilisateur et de code d'accès unique à l'application externe une fois les utilisateurs authentifiés. Cette interface est constituée de trois packs.

Les packs sont décrits dans le tableau ci-dessous.

Pack	Description
com.obi.ibm.um	Ce pack contient l'interface UMAPI (voir «UMAPI», à la page 460) qui fournit les méthodes d'implémentation d'un plug-in UM-SSO.
com.ibm.obi.um. exception	Ce pack définit l'exception générée par les méthodes UM-SSO, le cas échéant.
com.ibm.obi.um.vo	Ce pack contient les classes de création d'objets nécessaires aux méthodes de l'interface UMAPI, notamment les classes Données d'identification et Utilisateur.

Fonctionnement du plug-in UM-SSO

IBM Cognos Real-time Monitoring peut utiliser la même logique d'authentification que n'importe quelle autre application Web externe à l'aide de la structure du plug-in UM-SSO. Les tâches spécifiques à l'application externe sont implémentées dans le plug-in pendant que la structure vérifie que tous les accès protégés utilisent le plug-in pour l'authentification.

Les actions du plug-in UM-SSO peuvent être regroupées en quatre scénarios de base :

- Connexion initiale à Cognos Real-time Monitoring.
- Connexion à Cognos Real-time Monitoring et accès à l'application externe.
- Connexion à l'application externe et accès à Cognos Real-time Monitoring.
- Synchronisation des utilisateurs.

Connexion initiale à IBM Cognos Real-time Monitoring

Lorsqu'un utilisateur se connecte à IBM Cognos Real-time Monitoring, l'application externe appelle la méthode doAuthenticate à l'aide des données d'identification qui contiennent le nom d'utilisateur et le mot de passe de l'utilisateur. Si ces données sont valides, la méthode doAuthenticate(Credential, HttpRequest, HttpResponse) renvoie un objet SessionInfo utilisé par Cognos Real-time Monitoring pour maintenir la session utilisateur.

Pour plus d'informations, voir «doAuthenticate», à la page 460, «Données d'identification», à la page 464 et «SessionInfo», à la page 467.

Connexion à IBM Cognos Real-time Monitoring et accès à l'application externe

Dans ce scénario, la procédure de connexion est identique à celle de la connexion initiale. Cependant, la méthode doAuthenticate(Credential, HttpRequest,

© Copyright IBM Corp. 2007, 2014

HttpResponse) doit également définir des cookies et/ou des paramètres HTTP de sorte que l'application externe puisse les utiliser pour valider la session en cours.

Connexion à l'application externe et accès à IBM Cognos Real-time Monitoring

Dans ce scénario, la connexion à l'application externe doit ajouter un paramètre ou un cookie HTTP qui permet à l'application de valider ses propres sessions. Lorsque l'utilisateur accède à IBM Cognos Real-time Monitoring, ces cookies et ces paramètres restent dans la requête et deviennent disponibles pour la méthode doAuthenticate(HttpRequest, HttpResponse) du plug-in.

Dans ce scénario, le plug-in authentifie la session à l'aide de la même logique que celle utilisée par l'application externe. Si un objet SessionInfo (voir «SessionInfo», à la page 467) a été précédemment créé pour cette session, le même objet doit être renvoyé. Cela garantit la conservation des données Cognos Real-time Monitoring associées à cette session. Ce scénario est traité de la même façon que celui dans lequel l'utilisateur se connecte à Cognos Real-time Monitoring et accède à une autre page Cognos Real-time Monitoring. Si la méthode doAuthenticate ne parvient pas à authentifier l'utilisateur, il génère une exception de type. (Par exemple, il lance une exception SessionTimedOutException, car la session expire.) Pour en savoir davantage, voir «SessionTimedOutException», à la page 470.

Synchronisation des utilisateurs

IBM Cognos Real-time Monitoring planifie la synchronisation des utilisateurs à des intervalles réguliers. Lors de la synchronisation, Cognos Real-time Monitoring appelle les méthodes fetchRoles et fetchAllUsers dans le plug-in, qui doivent renvoyer l'ensemble des rôles et des utilisateurs autorisés à accéder à ce produit. La méthode fetchRoles(User) est appelée lorsqu'un utilisateur est authentifié, mais qu'il n'est pas synchronisé à Cognos Real-time Monitoring. Ceci permet à un utilisateur d'être synchronisé et d'établir sa connexion.

Installation d'un plug-in UM-SSO

Après avoir créé un plug-in UM-SSO à l'aide de l'interface UMAP, vous pouvez installer ce plug-in.

Procédure

1. Créez un fichier JAR qui comporte votre implémentation. Ce fichier doit contenir un fichier manifeste similaire à celui requis pour les fonctions UDF.

   Vous trouverez ci-dessous un exemple manifeste à inclure dans le fichier JAR.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"
?>
- <jarManifest xsi:schemaLocation="http://obi.cognos.com/4 jarManifest.xsd"
xmlns="http://obi.cognos.com/5" xmlns:xsi="http://www.w3.org/2001/
XMLSchemainstance">
  <jarManifestXSDVersion>1</jarManifestXSDVersion>
  <author>IBM Cognos Realtime Monitoring</author>
  <UmssoPlugin>
  <name>DummyPlugin</name>
  <description>Example plug-in</description>
  <implementor>com.ibm.umsso.example.ExamplePlugin</implementor>
  </UmssoPlugin>
  </jarManifest>
```

Pour plus d'informations sur les manifestes, voir «Fichiers manifestes», à la page 361.

L'élément `implementor` précise le nom du pack qui contient le plug-in.

2. Dans IBM Cognos Real-time Monitoring Workbench, cliquez sur **Plan de travail**.
3. Sous **Objets du plan de travail**, cliquez sur **Fonctions définies par l'utilisateur**.
4. Cliquez sur **Nouvelle fonction définie par l'utilisateur**.
5. Dans la fenêtre Sélectionner un fichier JAR, choisissez **Télécharger un fichier JAR**.
6. Saisissez un **Nom**, une **Description** et un **Chemin d'accès** pour le fichier JAR.
7. Cliquez sur **Continuer**.
8. Cliquez sur **Terminer** pour télécharger le fichier JAR.

Fichier XML setPropertyDetails

Un plug-in UM-SSO peut spécifier des détails de propriétés qui engendrent des paramètres supplémentaires dans les paramètres du système de la console d'administration d' IBM Cognos Real-time Monitoring Workbench.

Les détails de ces propriétés sont indiqués dans le fichier XML fourni par la méthode getSettings (voir «getSettings», à la page 464) à IBM Cognos Real-time Monitoring Workbench. Lorsque vous implémentez une méthode getSettings dans un plug-in, vous devez indiquer le fichier XML renvoyé par cette méthode sous la forme de chaîne. Le format de ce fichier XML est décrit dans les sections suivantes.

Eléments du fichier XML setPropertyDetails

L'élément de niveau supérieur des détails de propriété XML est setPropertyDetails. Il contient deux sous-éléments : Group et Property.

Elément de groupe

L'élément Group définit la page ajoutée par le plug-in UM-SSO aux paramètres système de la console d'administration d' IBM Cognos Real-time Monitoring Workbench. Un élément setPropertyDetails peut contenir un nombre illimité d'éléments Group et tous les éléments Group doivent figurer au début du document XML. L'élément Group contient deux sous-éléments : name et nameKey.

name

L'élément name indique le nom de groupe des propriétés relatives à l'élément Group. Tout élément Property qui utilise ce nom est lié au groupe. Par exemple, si un élément Group indique un élément name avec la valeur "Xyz", tout élément Property qui contient la valeur "Xyz" dans son sous-élément propertyGroup appartient au groupe Xyz.

nameKey

L'élément nameKey représente la clé du nom de groupe des propriétés correspondant à une chaîne localisée gérée dans un regroupement de ressources qui est renvoyé par la méthode getResourceBundle (voir «getResourceBundle», à la page 463) de l'interface UMAPI (voir «UMAPI», à la page 460). La chaîne du regroupement de ressources qui correspond à l'élément KeyName représente le nom du paramètre visible par l'utilisateur dans les **Paramètres du système**.

Elément de propriété

L'élément Property détermine les commandes qui s'affichent sur la page des paramètres système. Tous les éléments Property doivent correspondre aux éléments Group définis dans le document XML. Toutefois, plusieurs éléments Property peuvent faire référence au même élément Group.

Un élément Property comporte les sous-éléments suivants :

- name
- nameKey
- descriptionKey
- uiProperty

name

Le nom de l'élément Property agit comme un identificateur pouvant être utilisé par un implémenteur de plug-in UM-SSO pour rechercher la propriété.

nameKey

Le sous-élément nameKey représente la clé de la propriété correspondant à une chaîne localisée stockée dans un regroupement de ressources. Le regroupement de ressources est renvoyé par la méthode «getResourceBundle», à la page 463 de l'interface «UMAPI», à la page 460. La chaîne stockée dans le regroupement de ressources associée au nom clé correspond au nom du libellé affiché dans les **Paramètres du système** de cette propriété.

descriptionKey

Le sous-élément descriptionKey représente la clé qui correspond à une chaîne localisée stockée dans le regroupement de ressources permettant de fournir une description de la propriété. Le regroupement de ressources est renvoyé par la méthode «getResourceBundle», à la page 463 de l'«UMAPI», à la page 460. Le texte de la description apparaît sous forme d'infobulle lorsque l'utilisateur déplace le pointeur de la souris sur la propriété dans la page **Paramètres du système**.

uiProperty

Le sous-élément uiProperty associe l'élément Property à un élément Group, précise l'ordre dans lequel la propriété s'affiche dans la page **Paramètres du système** lorsque plusieurs propriétés sont associées à un groupe et indique si la propriété est obligatoire ou non.

L'élément uiProperty comporte les sous-élément suivants :

- propertyGroup

  Nom du groupe auquel appartient l'élément Property. Par exemple, si l'élément propertyGroup possède la valeur "Xyz", IBM Cognos Real-time Monitoring associe l'élément Property à l'élément Group qui s'intitule "Xyz".
- listOrder

  Si plusieurs éléments Property sont associés à un élément Group, listOrder précise l'ordre dans lequel les propriétés s'affichent sur la page **Paramètres du système**.
- isMandatory

Il s'agit d'un élément booléen. Sa valeur est définie sur False. Si vous définissez la valeur de l'élément isMandatory sur True, la propriété doit être définie lors de la première ouverture d' IBM Cognos Real-time Monitoring Workbench.

- defaultValue

  Indique la valeur par défaut de la propriété. Si vous spécifiez cet élément, vous ne pouvez pas définir d'élément defaultVariable.

- defaultVariable

  Indique la variable par défaut de la propriété. Si vous indiquez cet élément, vous ne pouvez pas définir d'élément defaultValue.

propertyTypeDefinition

L'élément propertyTypeDefinition détermine la commande relative à la propriété qui s'affiche sur la page **Paramètres du système**. Cette commande peut être une zone de texte ou une boîte de dialogue, une zone d'entier, un bouton radio, une case à cocher ou un panneau de planification.

Vous pouvez définir n'importe quel sous-éléments suivants de propertyTypeDefinition pour définir le type de commande à afficher :

- string

  Crée une zone de texte dans laquelle un utilisateur peut saisir une chaîne. Cet élément possède les attributs facultatifs suivants :

  – maxlength

    L'attribut maxLength indique la longueur maximale de la chaîne pouvant être saisie par l'utilisateur.

  – isPassword

    Indique que la zone de texte est réservée à la saisie d'un mot de passe. Un algorithme en texte clair est utilisé pour les mots de passe.

- multiString

  Crée une zone de texte dans laquelle l'utilisateur peut saisir plusieurs chaînes.

- int

  Indique une zone dans laquelle l'utilisateur peut saisir un entier. Cet élément possède deux attributs facultatifs :

  – min

    Spécifie la valeur entière minimale pouvant être saisie par l'utilisateur. Si les attributs min ne sont pas précisés, la valeur minimale est -2147483648

  – max

    Spécifie la valeur entière maximale pouvant être saisie par l'utilisateur. Si l'attribut max n'est pas précisé, la valeur maximale est 2147483647.

- enum

  Crée une liste déroulante sur la page **Paramètres du système**. Vous pouvez indiquer plusieurs éléments enum. Chaque élément enum possède les deux attributs suivants :

  – choiceKey

    Clé qui correspond à une chaîne localisée stockée dans un regroupement de ressources. La chaîne est utilisée dans le libellé associé au bouton radio. Le regroupement de ressources est renvoyé par la méthode getResourceBundle (voir «getResourceBundle», à la page 463) de l'interface UMAPI (voir «UMAPI», à la page 460).

  – choiceValue

Détermine la valeur renvoyée lorsque l'utilisateur sélectionne le bouton radio.

- Boolean

  Crée une case à cocher.

- Schedule

  Ajoute une interface de planification à la page qui permet à un utilisateur de sélectionner l'intervalle auquel planifier un événement, notamment une synchronisation.

UMAPI

UMAPI est une interface Java qui contient les méthodes implémentées dans un plug-in UM-SSO. Cette interface est fournie dans le pack com.obu.ibm.um.

doAuthenticate

Authentifie un utilisateur en fonction d'une requête et d'une réponse HTTP.

Un objet Credential (voir «Données d'identification», à la page 464) est inclus dans la méthode avec la requête et la réponse la première fois qu'un utilisateur se connecte ou que la session de l'utilisateur expire. Cette méthode est appelée lorsqu'IBM Cognos Real-time Monitoring veut effectuer une opération de connexion en mode d'accès unique. La méthode crée une session avec l'application externe.

Syntaxe

```
SessionInfo doAuthenticate(Credential credential, javax.servlet.http.
HttpServletRequest
request, javax.servlet.http.HttpServletResponse response)
```

```
SessionInfo doAuthenticate(javax.servlet.http.HttpServletRequest
request,javax.servlet.http.HttpServletResponse
response)
```

Paramètres

`credential`

Objet données d'identification qui contient le nom d'utilisateur et le mot de passe de l'utilisateur en cours d'authentification.

`request`

Objet HttpServletRequest qui contient les informations sur une requête HTTP, notamment l'authentification, les en-têtes et les cookies.

`response`

Objet HttpServletResponse qui contient les informations sur la réponse HTTP à la requête HTTP correspondante.

Valeur renvoyée

Informations de session relatives à la session en cours. L'objet SessionInfo contient un objet Utilisateur et les attributs associés.

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`InvalidCredentialException`

Si l'application externe a fourni des données d'identification incorrectes lorsqu'un utilisateur a tenté d'établir une connexion.

doLogout

Désactive la connexion utilisateur avec l'application externe. Cette méthode supprime la session en cours.

Syntaxe

```
void doLogout(SessionInfo sessionInfo, javax.servlet.http.HttpServletRequest
request, javax.servlet.http.HttpServletResponse response)
```

Paramètres

`sessionInfo`

Propriétés spécifiques à la session ainsi qu'à l'utilisateur pour lequel la session est maintenue.

`request`

Objet HttpServletRequest qui contient les informations relatives à la requête HTTP, par exemple, l'authentification, les en-têtes et les cookies.

`response`

Objet HttpServletResponse qui contient les informations relatives à la réponse correspondante.

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`InvalidCredentialException`

Si l'application externe a fourni des données d'identification incorrectes lorsqu'un utilisateur a tenté d'établir une connexion.

`SessionTimedOutException`

Si la session avec l'application externe a expiré.

fetchAllUsers

Extrait tous les utilisateurs capables d'interagir avec IBM Cognos Real-time Monitoring.

Syntaxe

```
java.util.Iterator fetchAllUsers(int batchSize)
```

Paramètre

`batchSize`

Nombre d'utilisateurs appropriés à extraire.

Valeur renvoyée

Itérateur doté d'objets de type Utilisateur (voir «Utilisateur», à la page 468). Le nombre d'objets de type Utilisateur renvoyés dépend de la valeur `batchSize`.

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`SessionTimedOutException`

Si la session avec l'application externe a expiré.

fetchRoles

Extrait tous les rôles capables d'interagir avec IBM Cognos Real-time Monitoring.

Syntaxe

```
public java.util.Iterator fetchRoles(int batchSize)
```

Paramètre

`batchSize`

Nombre maximal de rôles à renvoyer.

Valeur renvoyée

Itérateur doté d'objets de type Rôle (voir «Rôle», à la page 465). Le nombre maximal d'objets Rôle dépend de la valeur `batchSize`.

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`SessionTimedOutException`

Si la session avec l'application externe a expiré.

getResourceBundle

Extrait les chaînes localisées des attributs ou des clés utilisés par le plug-in. Ces chaînes sont renvoyées sous la forme d'un objet ResourceBundle.

Syntaxe

```
java.util.ResourceBundle getResourceBundle(java.util.Locale
locale)
```

Paramètre

`locale`

Nombre de rôles appropriés à extraire.

Valeur renvoyée

Objet ResourceBundle qui contient les chaînes localisées des attributs ou des clés utilisés dans le plug-in.

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`SessionTimedOutException`

Si la session avec l'application externe a expiré.

getRoles

Extrait les rôles affectés à l'utilisateur donné dans l'application externe.

Syntaxe

```
public java.util.Iterator getRoles(User user)
```

Paramètre

`user`

L'utilisateur dont les rôles sont extraits par la méthode.

Valeur renvoyée

Itérateur doté d'objets de type Rôle (voir «Rôle», à la page 465).

Exceptions

`ApplicationNotAvailableException`

Si l'application externe n'est pas disponible en raison d'un échec de connexion réseau, d'une maintenance système, etc.

`SessionTimedOutException`

Si la session avec l'application externe a expiré.

getSettings

Extrait les paramètres de configuration sous la forme d'une chaîne XML. Ces paramètres permettent à IBM Cognos Real-time Monitoring d'enregistrer le plug-in UM-SSO. L'implémentation de la méthode getSettings détermine la façon dont le format XML est extrait. Par exemple, le code XML peut être extrait par la méthode à partir d'un fichier texte géré par l'application externe ou être codé en dur dans la méthode.

Syntaxe

```
public java.lang.String getSettings()
```

Valeur renvoyée

Renvoie un document XML qui indique les paramètres de configuration.

onPropertyChange

La méthode onPropertyChange est une méthode de rappel à partir d' IBM Cognos Real-time Monitoring permettant d'informer le plug-in de toute modification éventuelle de propriétés par l'utilisateur dans les paramètres d'administration de Real-time Monitoring. Pour récupérer la valeur d'une propriété, le plug-in peut appeler : VCSystem.getVCProperty().getProperty(propertyName) ou VCSystem.getVCProperty().isPropertySet(propertyName). Par exemple, si un utilisateur modifie la propriété d'un calendrier au cours de la synchronisation, cette méthode arrête le processus de synchronisation.

Syntaxe

```
public void onPropertyChange(java.lang.String propertyName)
```

Paramètre

`propertyName`

Nom de la propriété qui a été modifiée dans Cognos Real-time Monitoring.

Données d'identification

La classe des données d'identification permet de créer des objets de valeur de type Credential qui contiennent les données nécessaires à l'authentification (nom d'utilisateur et mot de passe).

Un objet de valeur Credential est utilisé par la méthode doAuthenticate (voir «doAuthenticate», à la page 460) lorsque l'application externe doit connecter un utilisateur pour la première fois ou lorsqu'elle doit rétablir la session d'un utilisateur.

getPassword

Extrait le mot de passe contenu dans les données d'identification associées au nom d'utilisateur correspondant.

Syntaxe

```
public java.lang.String getPassword()
```

Valeur renvoyée

Mot de passe renvoyé sous la forme d'une chaîne.

getUserName

Extrait le nom d'utilisateur indiqué dans les données d'identification.

Syntaxe

```
public java.lang.String getUserName()
```

Valeur renvoyée

Nom d'utilisateur renvoyé sous la forme d'une chaîne.

setPassword

Définit le mot de passe dans les données d'identification.

Syntaxe

```
public void setPassword(java.lang.String password)
```

Paramètre

`password`

Mot de passe associé au nom d'utilisateur défini dans les données d'identification.

setUserName

Définit le nom d'utilisateur dans les données d'identification.

Syntaxe

```
public void setUserNmae(java.lang.String userName)
```

Paramètre

`password`

Nom d'utilisateur indiqué dans les données d'identification.

Rôle

La classe Rôle permet de créer des objets de valeur Rôle qui contiennent les attributs associés à un rôle ainsi que les utilisateurs appartenant à ce rôle.

getAttributes

Extrait les attributs spécifiés dans le rôle.

Syntaxe

```
public java.util.Map getAttributes()
```

Valeur renvoyée

Attributs du rôle renvoyés en tant que paires clé-valeur contenues dans un objet de mappage.

getRoleDescription

Extrait la description d'un rôle.

Syntaxe

```
public java.lang.String getRoleDescription()
```

Valeur renvoyée

Une chaîne qui contient la description du rôle.

getRoleName

Extrait le nom du rôle.

Syntaxe

```
public java.lang.String getRoleName()
```

Valeur renvoyée

Une chaîne qui contient le nom du rôle.

getUsers

Extrait les utilisateurs associés au rôle.

Syntaxe

```
public java.util.Iterator getUsers()
```

Valeur renvoyée

Un itérateur avec des objets de type Utilisateur (voir «Utilisateur», à la page 468).

setAttributes

Définit les attributs associés au rôle.

Syntaxe

```
public void setAttributes(java.util.Map attributes)
```

Paramètre

`attributes`

Attributs associés au rôle.

setRoleDescription

Définit la description du rôle.

Syntaxe

```
public void setRoleDescription(java.lang.String
roleDescription)
```

Paramètre

`roleDescription`

Description du rôle.

setRoleName

Définit le nom du rôle.

Syntaxe

```
public void setRoleName(java.lang.String roleName)
```

Paramètre

`roleName`

Nom du rôle.

setUsers

Définit les utilisateurs associés au rôle.

Syntaxe

```
public void setUsers(java.util.Iterator userIterator)
```

Paramètre

`userIternator`

Itérateur doté d'objets de type Utilisateur (voir «Utilisateur», à la page 468).

SessionInfo

La classe SessionInfo permet de créer des objets de valeur SessionInfo qui renferment les propriétés spécifiques à la session ainsi qu'à l'utilisateur pour lequel la session est maintenue. Une application externe peut utiliser des objets SessionInfo afin d'effectuer le suivi de la session en cours.

getAttributes

Extrait les attributs de la session.

Syntaxe

```
public java.util.Map getAttributes()
```

Valeur renvoyée

Renvoie un objet Map qui contient les attributs de la session.

getUser

Renvoie l'utilisateur associé à la session.

Syntaxe

```
public User getUser()
```

Valeur renvoyée

Renvoie un objet «Utilisateur».

setAttributes

Définit les attributs de la session SessionInfo.

Syntaxe

```
public void setAttributes(java.util.Map attributes)
```

Paramètre

`attributes`

Objet Map qui contient les attributs de la session sous forme de paires clé-valeur.

setUser

Définit l'utilisateur associé à l'objet SessionInfo.

Syntaxe

```
public void setUser(User user)
```

Paramètre

`user`

«Utilisateur» à associer à la session.

Utilisateur

La classe Utilisateur permet de créer un objet de valeur Utilisateur qui comporte les attributs associés à un utilisateur.

getAttributes

Extrait les attributs de l'utilisateur.

Syntaxe

```
public java.util.Map getAttributes()
```

Valeur renvoyée

Renvoie un objet Map qui contient les attributs de l'utilisateur en tant que paires clé-valeur.

getUserDescription

Extrait la description de l'utilisateur.

Syntaxe

```
public java.lang.String getUserDescription()
```

Valeur renvoyée

Renvoie la description de l'utilisateur en tant que chaîne.

getUserId

EXtrait l'ID de l'utilisateur.

Syntaxe

```
public java.lang.String getUserId()
```

Valeur renvoyée

Renvoie l'ID de l'utilisateur en tant que chaîne.

setAttributes

Définit les attributs de l'utilisateur.

Syntaxe

```
public void setAttributes(java.util.Map attributes)
```

Paramètre

`attributes`

Renvoie un objet Map qui contient les attributs de l'utilisateur en tant que paires clé-valeur.

setUserDescription

Définit la description de l'utilisateur.

Syntaxe

```
public void setUserDescription(java.lang.String
userDescription)
```

Paramètre

`userDescription`

Description de l'utilisateur.

setUserId

Définit l'ID de l'utilisateur.

Syntaxe

```
public void setUserId(java.lang.String userId)
```

Paramètre

`userId`

Renvoie l'ID de l'utilisateur.

Exceptions

L'API UM-SSO définit les trois exceptions qui peuvent être générées par un plug-in.

- ApplicationNotAvailableException
- InvalidCredentialException
- SessionTimedOutException

Ces exceptions sont indiquées dans le pack com.ibm.obi.um.exception.

ApplicationNotAvailableException

Le plug-in UM-SSO renvoie cette exception si l'application externe n'est pas disponible pour une raison quelconque, telle qu'un échec de la connexion réseau ou la maintenance du système.

InvalidCredentialException

Le plug-in UM-SSO génère cette exception lorsque l'application externe envoie des données d'identification non valides.

SessionTimedOutException

Le plug-in UM-SSO génère cette exception lorsque la session avec l'application externe expire.

Remarques

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

© Copyright IBM Corp. 2007, 2014

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 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 permettent, en tant qu'utilisateur, de collecter des informations identifiant les utilisateurs finals via les cookies ou d'autres technologies, nous vous recommandons de consulter les avis juridiques s'appliquant à une telle collecte de données, y compris les mentions légales de notification et d'accord.

Pour plus d'informations sur l'utilisation des diverses technologies, y compris les cookies, à ces fins, voir la politique de confidentialité d'IBM à l'adresse suivante : http://www.ibm.com/privacy et la déclaration de confidentialité en ligne d'IBM à l'adresse suivante : http://www.ibm.com/privacy/details dans la section "Cookies, balises Web et autres technologies" et "Déclaration de confidentialité des produits logiciels et des logiciels en tant que services (SaaS) d'IBM" à l'adresse suivante :http://www.ibm.com/software/info/product-privacy.

Marques

IBM, le logo IBM et 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 « Copyright and trademark information » à l'adresse www.ibm.com/legal/copytrade.shtml.

Les termes suivants 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 des définitions relatifs à IBM Cognos Real-time Monitoring.

Les références croisées suivantes sont utilisées :

- Voir fait référence à un synonyme préféré, un acronyme ou une abréviation de la forme développée.
- Voir aussi fait référence à un terme similaire ou contraire.

Pour consulter les glossaires d'autres produits IBM, accédez à www.ibm.com/software/globalization/terminology (s'ouvre dans une nouvelle fenêtre).

«A» «C» «D» «E» «F» «J», à la page 476 «M», à la page 476 «N», à la page 476 «P», à la page 476 «S», à la page 476 «T», à la page 476 «U», à la page 476 «V», à la page 476

A

activé
Objet acceptant et traitant de nouvelles données. Tous les objets sont créés avec l'état activé.

C

cascade
Processus propageant la même opération à tous les objets dépendants.

D

désactivé
Objet n'acceptant pas de nouvelles données. La désactivation d'un objet n'affecte pas sa définition ou son existence, mais empêche les nouvelles données de circuler vers cet objet et tous ceux qui dépendent de l'objet cible.

E

événement
Ligne ou série de lignes de données.

expression scalaire
Expression dépourvue d'une fonction d'ensembles (Set).

F

fenêtre de requête
Indique un ensemble de lignes utilisées dans les calculs par rapport à la ligne examinée. Le calcul peut porter sur une fonction d'ensembles mobiles (Moving set), sur une jointure ou sur les lignes arrivant à expiration d'une vue.

fenêtre en défilement continu
Fenêtre qui se vide de son contenu lorsqu'elle avance pour inclure le tout dernier événement.

flux de données
Objet qui représente un événement.

flux de données consolidé
Table de flux de données capturant des flux de données de sources distinctes mais similaires pour les combiner en un seul flux de données.

fonction de classement (Rank)
Fonction calculant le résultat scalaire de chaque valeur d'un ensemble par rapport à la totalité de l'ensemble. Une fonction de classement est uniquement utilisable dans la liste de sélection d'une instruction SELECT.

fonction de classement terminale
Fonction de classement (Rank) ne comportant que des arguments scalaires.

fonction définie par l'utilisateur (UDF)
Fonction vous permettant d'étendre C-SQL en définissant vos propres fonctions à utiliser dans les formules, y compris dans les requêtes, les expressions de zone et les règles.

fonction d'ensembles en défilement continu (Tumbling set)
Fonction effectuant des calculs sur un ensemble fenêtré de lignes d'une vue. L'ensemble de lignes à inclure est uniquement déterminé à l'arrivée d'un nouvel événement et se vide lorsqu'il est plein.

fonction d'ensembles mobiles (Moving set)
Fonction effectuant des calculs sur un ensemble de lignes les plus récentes d'une

© Copyright IBM Corp. 2007, 2014

vue. L'ensemble de lignes à inclure est uniquement déterminé à l'arrivée d'un nouveau flux de données.

fonction d'ensembles (Set)
Fonction effectuant des calculs sur une colonne dans un ensemble de lignes d'une vue, tels que le coût moyen de plusieurs commandes de produits similaires. Une fonction d'ensembles est uniquement utilisable dans la liste de sélection d'une instruction SELECT.

fonction d'ensembles terminale
Fonction d'ensembles (Set) ne comportant que des arguments scalaires.

fonction scalaire (Scalar)
Fonction SQL qui accepte des arguments en option et qui renvoie une valeur scalaire unique chaque fois qu'elle est appelée. Ces fonctions peuvent apparaître dans n'importe quelle expression C-SQL valide.

J

jointure externe
Jointure dont le résultat est constitué des lignes correspondantes dans les deux tables qui ont été reliées et des lignes qui n'ont eu aucune correspondance dans l'une ou l'autre des tables, ou les deux. Voir aussi jointure interne.

jointure interne
Résultat d'une opération de jointure qui inclut uniquement les lignes correspondantes dans les deux tables reliées. Voir aussi jointure externe.

M

modelage des données métier
Technique de description des événements, du contexte, des vues et des règles qui synthétisent le fonctionnement de votre activité.

N

non valide
Objet comportant une référence à un autre objet qui ne peut être satisfaite. Une référence peut être non valide lorsqu'un objet n'existe pas ou si certains attributs d'un objet ne correspondent pas aux exigences du dépendant (tel qu'un type de données ne correspondant pas), mais pas lorsque le dépendant est désactivé. Notez que tous les objets qui dépendent d'un objet non valide sont également non valides.

P

profil de diffusion
Indique où et comment envoyer des alertes et des données à l'utilisateur.

S

scénario
Séquence d'événements permettant de rechercher les résultats attendus ou potentiels de modèles de données métier et d'identifier des conditions métier exceptionnelles.

T

table virtuelle
Table ou vue dont les lignes sont dérivées à mesure qu'elles sont requises en mémoire.

U

UDF Voir fonction définie par l'utilisateur.

V

vue avec état
Vue contenant les résultats des agrégations dérivées des flux de données passés d'une seule ligne. Une vue est dite avec état si elle contient une fonction d'ensembles (Set) ou une fonction d'ensembles mobiles (Moving set) dans la clause SELECT, si elle contient une clause GROUP BY (auquel cas il existe une ligne par groupe) ou si elle est dérivée d'une vue avec état.

vue instantanée
Réplique d'une vue au moment de l'exécution de la requête.

vue récente
Copie instantanée de la dernière vue en cours non vide. Une vue récente correspond à ce qui apparaît dans

Workbench lors de la modification d'un objet qui affiche les résultats de la vue.

vue sans état
Vue qui contient uniquement les lignes représentant l'effet du dernier événement.