

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 477.

Première édition - juillet 2009

Réf. US : SC19-1020-02

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.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2009. Tous droits réservés.

© **Copyright International Business Machines Corporation 1998, 2009.**

Table des matières

Avis aux lecteurs canadiens. xi

Chapitre 1. Systèmes fédérés 1

Sources de données prises en charge	2
Le serveur fédéré.	5
Qu'est-ce qu'une source de données ?	5
La base de données fédérée	6
Encapsuleurs et modules d'encapsuleur	6
Comment interagir avec un système fédéré	7
Centre de contrôle DB2.	8
Programmes d'application	9
Outils de la famille DB2	9
IBM Optim Data Studio	9
Fournisseurs de services Web	10
Catalogue système des bases de données fédérées	10
Le compilateur SQL	11
L'optimiseur de requêtes	11
Compensation	12
Sessions passe-système	13
Définitions et options de serveur	13
Mappages utilisateur	14
Pseudonymes et objets de source de données	15
Objets de source de données valides	16
Options de colonne de pseudonymes.	16
Mappages de type de données	17
Mappages de fonction	18
Spécifications de l'index	18
Procédures mémorisées fédérées	19
Séquences de classement	19
Comment les séquences de classement	
déterminent les ordres de tri.	19
Définition de la séquence de classement locale	
pour optimiser les requêtes	20

Chapitre 2. Modification des configurations de sources de données . 23

Modification d'un encapsuleur (Centre de contrôle DB2).	23
Modification d'un encapsuleur - exemples	24
Modification d'un encapsuleur (ligne de commande DB2).	24
Modification des définitions et des options de serveur	25
Restrictions applicables à la modification de définitions de serveur	26
Modification de la version de la source de données dans une définition de serveur (Centre de contrôle DB2).	27
Modification de la version de la source de données dans une définition de serveur (ligne de commande DB2).	27
Modification de toutes les définitions de serveur pour un type de source de données particulier	28
Utilisation d'options de serveur dans des définitions de serveur (Centre de contrôle DB2)	28

Modification temporaire des options de serveur pour les sources de données relationnelles	29
Hiérarchie des paramètres des options de serveur	30
Utilisation d'options de serveur dans des définitions de serveur (ligne de commande DB2).	30
Modification d'un mappage utilisateur (Centre de contrôle DB2).	31
Modification d'un mappage utilisateur (ligne de commande DB2).	32
Modification d'un pseudonyme (Centre de contrôle DB2).	33
Restrictions pour les modifications de pseudonymes.	34
Modification des noms de colonne de pseudonymes (Centre de contrôle DB2)	35
Modification des noms de colonne de pseudonymes (ligne de commande DB2)	36
Modification des options de pseudonymes (Centre de contrôle DB2)	37
Modification des options de pseudonymes (ligne de commande DB2).	38
Modification des options de colonne de pseudonymes (Centre de contrôle DB2)	39
Modification des options de colonne de pseudonymes (ligne de commande DB2)	40
Modification d'un pseudonyme (ligne de commande DB2).	41
Suppression d'un encapsuleur	42
Suppression d'une définition de serveur.	43
Suppression d'un mappage utilisateur	44
Suppression d'un pseudonyme	45

Chapitre 3. Mappages de types de données dans un système fédéré . . . 47

Mappages de types de données et catalogue global de bases de données fédérées	48
Quand créer des mappages de types de données alternatifs	49
Data type mappings for nonrelational data sources	49
Mappages de types de données avals et inversés	49
Création de mappages de types de données	50
Création d'un mappage de type pour un type de données de source de données - exemple	51
Création d'un mappage de type pour un type de données et une version de source de données - exemple	52
Création d'un mappage de type pour tous les objets de source de données sur un serveur - exemple	53
Transtypage entre types de données	54
Prise en charge du type de données TIMESTAMP	55
Modification d'un type local pour un objet de source de données (Centre de contrôle DB2)	55
Modification d'un type local d'objet de source de données - exemples	56

Modification d'un type local pour un objet de source de données (ligne de commande DB2) . . .	58
Modification des types de données LONG en types de données VARCHAR	59

Chapitre 4. Mappage de fonctions et de fonctions définies par l'utilisateur . . . 61

Mappages de fonctions dans un système fédéré . . .	61
Principes de fonctionnement des mappages dans un système fédéré	61
Pourquoi les mappages de fonction sont importants.	62
Quand créer les mappages de fonction	62
Fonctions définies par l'utilisateur dans les applications	63
Exigences pour le mappage de fonctions définies par l'utilisateur	63
Création de mappages de fonction.	64
Spécification de noms de fonction dans l'instruction CREATE FUNCTION MAPPING . . .	64
Création d'un mappage de fonction pour un type de source de données spécifique	65
Création d'un mappage de fonction pour une version et un type de source de données spécifiques	66
Création d'un mappage de fonction pour tous les objets de source de données sur un serveur spécifique	66
Modèles de fonctions	67
Création de modèles de fonction	67
Désactivation d'un mappage de fonction par défaut	69
Suppression d'un mappage de fonction	70

Chapitre 5. Création de spécifications d'index. 71

Spécifications d'index dans un système fédéré. . .	71
Création des spécifications d'index pour les objets de source de données	72
Création de spécifications d'index sur les tables acquérant de nouveaux index	73
Création de spécifications d'index sur les vues . . .	74
Création de spécifications d'index sur les synonymes Informix	76

Chapitre 6. Développement de procédures fédérées. 79

Procédures fédérées	79
Sources de données et procédures fédérées	81
Procédures fédérées pour les sources de données DB2	81
Procédures fédérées et Microsoft SQL Server . . .	82
Procédures fédérées et Oracle	84
Procédures fédérées et Sybase	88
Création de procédures fédérées	90
Octroi ou révocation d'autorisations d'appel de procédures fédérées	91
Affichage des informations relatives à un paramètre et appel de procédures fédérées	93
Modification ou suppression de procédures fédérées	94

Jointure d'ensembles de résultats pour les procédures fédérées	95
Commande DB2FEDGENTF.	98
Identification et résolution d'incidents dans une procédure fédérée	99

Chapitre 7. Création et modification de tables distantes à l'aide de DLL transparent 103

Qu'est-ce que le DDL transparent ?	103
Colonnes LOB distantes et DDL transparent . . .	104
Création de tables distantes et de DDL transparent	104
Création de tables distantes à l'aide de DDL transparent	105
Création de tables distantes à l'aide de DDL transparent - exemples	106
Modification de tables distantes à l'aide de DDL transparent	108
Suppression de tables distantes à l'aide de DDL transparent	109

Chapitre 8. Gestion de transactions dans un système fédéré. 111

Description du support des transactions du système fédéré	111
Mise à jour dans un système fédéré	113
Transaction de mise à jour dans une session passe-système	114
Source de données validant de manière automatique des instructions DDL	114
Fonctions définies par l'utilisateur transmises à la source de données pour traitement	114

Chapitre 9. Transactions de validation en deux phases 115

Validation en deux phases pour les transactions fédérées	115
Planification pour la validation en deux phases fédérées	116
Architecture fédérée pour la validation en deux phases.	116
Validation en deux phases pour les transactions fédérées - Exemples	118
Procédure de traitement des transactions de validation en deux phases	122
Activation d'une validation en deux phases pour les transactions fédérées.	125
Configuration requise pour la source de données et pour les transactions de validation en deux phases fédérées	127
Configuration des sources de données DRDA	128
Configuration des sources de données Oracle	128
Configuration des sources de données Informix	129
Configuration des sources de données Microsoft SQL Server	130
Configuration des sources de données Sybase	131
Reprise suite à des incidents de validation en deux phases fédérées.	132
Resynchronisation pour les systèmes fédérés	132

Reprise manuelle de transactions en attente de validation	133
Traçage des états des transactions d'unité d'oeuvre répartie dans des sources de données .	134
Résolution des incidents de validation en deux phases fédérées.	136
Performances de la validation en deux phases fédérées	137
Amélioration des performances de validation en deux phases fédérées.	138

Chapitre 10. Insertion, mise à jour et suppression des données dans un système fédéré 139

Privilèges des autorisations pour les instructions INSERT, UPDATE et DELETE	139
Restrictions INSERT, UPDATE et DELETE du système fédéré	140
Sources de données non prises en charge	140
Intégrité référentielle dans un système fédéré.	140
Instructions INSERT, UPDATE et DELETE et objets LOB	141
Conservation de l'atomicité des instructions dans un système fédéré	141
Modification de données dans un système fédéré	142
Insertion de données dans des objets de source de données	142
Mise à jour des données dans les sources de données	142
Suppression de données des objets de source de données	143
Sémantiques d'attribution dans un système fédéré	144
Sémantiques d'attribution dans un système fédéré - Exemples	146
Restrictions des sources de données sur des valeurs de types de données	146
Mises à jour fédérées à l'aide des points de sauvegarde d'une application	148
Exemples de mises à jour fédérées à l'aide des points de sauvegarde d'une application	148
Restrictions appliquées aux les mises à jour fédérées à l'aide des points de sauvegarde d'une application	149

Chapitre 11. Importation et exportation de données pour les pseudonymes . . 151

Restrictions pour l'importation des données dans des pseudonymes	151
Commande IMPORT avec des pseudonymes - Exemples.	152
Restrictions pour l'exportation de données à l'aide de pseudonymes	153

Chapitre 12. Utilisation de pseudonymes 155

Pseudonyme dans un système fédéré	155
Curseurs déclarés WITH HOLD	156
Déclencheurs	156
Accès aux données avec des pseudonymes	156

Instructions SQL pouvant être utilisées avec des pseudonymes	156
Accès aux nouveaux objets de source de données	160
Création de pseudonymes pour les sources de données relationnelles et non relationnelles	161
Noms d'index et de colonnes de pseudonymes	162
Accès aux sources de données à l'aide de sessions passe-système	163
Accès aux données hétérogènes via des vues fédérées	163
Création de vues fédérées - Exemples	164
Création d'un pseudonyme sur un pseudonyme	165
Sélection de données dans un système fédéré.	166
Sélection de données dans un système fédéré - Exemples.	166
Contraintes informationnelles sur les pseudonymes	169
Indication des contraintes informationnelles sur les pseudonymes (Centre de contrôle DB2)	169
Indication des contraintes informationnelles sur les pseudonymes (ligne de commande DB2)	170
Spécification de contraintes informationnelles sur des pseudonymes - Exemples.	171
Mise à jour des statistiques de pseudonyme	174
Fonction de mise à jour des statistiques de pseudonymes - présentation	174
Méthodes d'extraction des statistiques de pseudonyme	176
Extraction de statistiques de pseudonyme	176
Restrictions relatives aux statistiques HIGH2KEY et LOW2KEY	180
Création d'un catalogue d'outils DB2	181
Affichage de l'état des mises à niveau des statistiques de pseudonyme (Centre de contrôle DB2)	181
Affichage de l'état des mises à niveau des statistiques de pseudonyme (ligne de commande DB2)	181
Procédure mémorisée SYSPROC.NNSTAT	182
Mise à jour automatique des statistiques de pseudonymes	184

Chapitre 13. Utilisation des données XML distantes 187

Exemples de création d'un alias pour les données XML distantes	187
Exemples de manipulation des données XML	187
Validation de documents XML à l'aide de schémas XML	188
Enregistrement des schémas XML	188
Validation des documents XML	189
Décomposition des documents XML à l'aide des schémas XML annotés dans les alias.	190
Traitement fédéré des données XML distantes	190
Analyse syntaxique fédérée des données XML distantes	190
Emissions des pages de codes pour les données XML distantes	191
Restrictions appliquées au type de données XML distantes	191

Chapitre 14. Tolérances des erreurs dans les expressions de tables imbriquées	193
Spécification pour les expressions de tables imbriquées pour la tolérance aux erreurs	194
Expressions de tables imbriquées pour la tolérance aux erreurs - Exemple	195
Support de source de données pour les expressions de tables imbriquées pour la tolérance aux erreurs	196
Restrictions applicables aux expressions de tables imbriquées pour la tolérances aux erreurs	196
Chapitre 15. Surveillance des serveurs et pseudonymes fédérés	197
Indicateurs de santé pour les pseudonymes et serveurs fédérés	197
Activation des indicateurs de santé fédérés	198
Surveillance de la santé des pseudonymes et serveurs fédérés	198
Surveillance de la santé des pseudonymes et serveurs fédérés - exemple	199
Surveillance d'instantané des systèmes fédérés - Présentation	200
Surveillance des requêtes fédérées	200
Surveillance d'instantané des requêtes fédérées - exemple	202
Eléments du moniteur des systèmes de base de données fédérée	203
Chapitre 16. Mode d'interaction des applications client avec les sources de données	205
Chapitre 17. Référence aux objets de source de données à l'aide de pseudonymes dans des instructions SQL	207
Pseudonymes dans les instructions DDL	207
Impact des statistiques de sources de données sur les applications	208
Définition des options de colonne sur les pseudonymes	209
Définition de l'option de colonne NUMERIC_STRING	210
Définition de l'option de colonne VARCHAR_NO_TRAILING_BLANKS	210
Chapitre 18. Création et utilisation des vues fédérées	211
Création de vues fédérées - Exemples	212
Chapitre 19. Préserver l'intégrité des données avec les niveaux d'isolement	213
Isolement au niveau de l'instruction dans un système fédéré	214
Isolement au niveau de la connexion dans un système fédéré	215

Chapitre 20. Prise en charge des objets LOB fédérés	217
Releveurs de coordonnées LOB	218
Restrictions sur les objets LOB	218
Considérations de performances pour le traitement des objets LOB	219
Chapitre 21. Requêtes réparties pour effectuer des requêtes sur des sources de données	221
Requêtes réparties pour effectuer des requêtes sur des sources de données - exemples	221
Optimisation des requêtes réparties avec des options de serveur	223
Annulation d'une requête fédérée	224
Chapitre 22. Interrogation des sources de données avec des sessions passe-système	227
Considérations et restrictions relatives au passe-système fédéré	228
Sessions passe-système vers des sources de données Oracle	230
Chapitre 23. Réglage fin du traitement des requêtes	231
Publications sur les performances des données fédérées	231
Analyse de requête	232
Analyse du transfert de l'exécution sur la base de données	234
Caractéristiques de serveur ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données	235
Différences SQL	236
Séquence de classement	238
Options de serveur fédéré	240
Facteurs de mappage de type et de fonction	241
Caractéristiques de pseudonymes ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données	242
Type de données locales d'une colonne de pseudonyme	242
Options de colonne fédérée	243
Caractéristiques de requête ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données	244
Analyse de l'endroit où une requête est évaluée	244
Analyse de l'endroit où une requête est évaluée avec l'option de serveur DB2_MAXIMAL_PUSHDOWN	245
Présentation des décisions d'évaluation de plan d'accès	245
Pourquoi ce prédicat n'est-il pas évalué à distance ?	245
Pourquoi l'opérateur GROUP BY n'est-il pas évalué à distance ?	246
Pourquoi l'opérateur SET n'est-il pas évalué à distance ?	246

Pourquoi l'opération ORDER BY n'est-elle pas évaluée à distance ?	247
Pourquoi une instruction INSERT éloignée avec une instruction FULLSELECT n'est-elle pas intégralement évaluée à distance ?	247
Pourquoi une instruction INSERT éloignée avec une clause VALUES n'est-elle pas intégralement évaluée à distance ?	247
Pourquoi une instruction UPDATE recherchée à distance n'est-elle pas intégralement évaluée à distance ?	248
Pourquoi une instruction UPDATE de position n'est-elle pas intégralement évaluée ?	248
Pourquoi une instruction DELETE de recherche, éloignée, n'est-elle pas intégralement évaluée à distance ?	248
Mises à niveau et personnalisation des sources de données	249
Transfert de l'exécution sur la base de données des prédicats avec le modèle de fonction	249

Chapitre 24. Parallélisme avec des requêtes qui font référence aux pseudonymes 251

Parallélisme intrapartition avec des requêtes qui font référence aux pseudonymes	251
Activation du parallélisme intrapartition avec les requêtes qui référence des pseudonymes	251
Parallélisme intrapartition avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès	252
Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes	254
Activation du parallélisme interpartition avec les requêtes qui font référence à des pseudonymes	257
Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès	258
Groupes de partitions calculés	260
Définition d'un groupe de partitions calculé	261
Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes - Attentes en matière de rendement	262
Parallélisme mixte avec des requêtes qui font référence aux pseudonymes	262
Activation du parallélisme mixte avec les requêtes qui font référence à des pseudonymes	262
Parallélisme mixte avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès	263

Chapitre 25. Traitement asynchrone des requêtes fédérées 265

Traitement asynchrone des requêtes fédérées - Exemples	265
Optimisation de l'état asynchrone	266
Plans d'accès sans état asynchrone	266
Plans d'accès optimisés pour l'état asynchrone	266
Plans d'accès - Exemples	267

Contrôle de la consommation des ressources	271
Activation de l'optimisation de l'état asynchrone	271
Remarques sur le réglage de l'optimisation de l'état asynchrone	272
Restrictions relatives à l'optimisation de l'état asynchrone	273
Indication de l'application ou non de l'optimisation de l'état asynchrone à une requête	273

Chapitre 26. Optimisation globale. . . 275

Caractéristiques de serveur ayant une incidence sur l'optimisation globale.	275
Rapport relatif de la vitesse de l'unité centrale	276
Rapport relatif de la vitesse d'entrée-sortie	277
Débit de communication entre le serveur fédéré et la source de données	277
Séquences de classement de source de données	277
Indications de plan éloigné	278
Caractéristiques des pseudonymes ayant une incidence sur l'optimisation globale	278
Spécifications de l'index	278
Statistiques du catalogue global	279
Mise à jour des modifications apportées aux lignes	280
Mise à jour des statistiques lors de modifications apportées aux colonnes	281
Analyse de l'optimisation globale	281
Présentation des décisions d'optimisation de plan d'accès	282
Pourquoi une jointure entre deux pseudonymes d'une même source de données n'est-elle pas évaluée à distance ?	282
Pourquoi l'opérateur GROUP BY n'est-il pas évalué à distance ?	282
Pourquoi l'instruction n'est-elle pas intégralement évaluée à distance ?	283
Pourquoi un plan généré par l'optimiseur et intégralement évalué à distance obtient-il de moins bonnes performances que la requête originale exécutée directement sur la source de données éloignée ?	283

Chapitre 27. Éléments du contrôleur système qui ont une incidence sur les performances 285

Chapitre 28. Tables de requêtes matérialisées 287

Tables de requêtes matérialisées et systèmes fédérés - Généralités	287
Création d'une table de requêtes matérialisées fédérées	288
Restrictions propres aux sources de données des tables de requêtes matérialisées	288
Restrictions sur l'utilisation des tables de requêtes matérialisées avec les pseudonymes	289

Chapitre 29. Tables de la mémoire cache. 291

Création de tables de mémoire cache	293
Modification des paramètres pour les tables de requêtes matérialisées	294
Ajout de tables de requêtes matérialisées à une table de mémoire cache	294
Routage de requêtes vers des tables de mémoire cache	295
Activation et désactivation des paramètres du cache de réplication	296
Suppression de tables de requêtes matérialisées d'une table de mémoire cache	297
Suppression de tables de mémoire cache	297

Chapitre 30. Sécurité pour les serveurs fédérés 299

Chapitre 31. Contextes accrédités fédérés et connexions accréditées . . . 301

Avantages des connexions accréditées fédérées	302
Types de connexions accréditées fédérées	303
API pour les connexions accréditées fédérées	304
Scénarios pour l'implémentation de connexions accréditées fédérées	305
Scénario : connexions accréditées fédérées de bout en bout, sans mappages utilisateur	305
Code échantillon pour les scénarios de connexion accréditée fédérée de bout en bout	308
Scénario : connexions accréditées fédérées de bout en bout, avec mappages utilisateur	310
Scénario : Connexions accréditées sortantes fédérées	313
Mappages utilisateurs et connexions accréditées fédérées	318

Chapitre 32. Contrôle d'accès basé sur les intitulés (LBAC) et systèmes fédérés 321

Chapitre 33. Référentiels de mappage utilisateur externes 323

Plug-in de mappage utilisateur (langage de programmation C).	323
Plates-formes prises en charge pour le plug-in de mappage utilisateur (langage de programmation C).	324
Restrictions au développement d'un plug-in de mappage utilisateur (langage de programmation C)	325
Fichier d'en-tête (fsumplugin.h) pour le plug-in de mappage utilisateur (langage de programmation C).	326
Développement d'un plug-in de mappage utilisateur (langage de programmation C)	330
Echantillon de plug-in de mappage utilisateur (en langage de programmation C)	335
Plug-in de mappage utilisateur (langage de programmation Java).	336
Classes de plug-in de mappage utilisateur (langage de programmation Java)	337

Plug-in exemple de mappage utilisateur (langage de programmation Java)	342
Développement d'un plug-in de mappage utilisateur (langage de programmation Java)	343

Chapitre 34. Sécurité Oracle dans un système fédéré 351

Oracle Label Security	351
Authentification de proxy Oracle et contextes accrédités fédérés	352

Chapitre 35. Support de source de données pour des fonctions fédérées . 353

Chapitre 36. Référence des options des sources de données 357

Référence des options BioRS	357
Référence des options de base de données DB2	361
Référence des options Excel	369
Référence des options Informix	370
Référence des options JDBC	375
Référence des options Microsoft SQL Server	382
Référence des options ODBC	388
Référence des options Oracle	393
Référence des options de script	398
Référence des options Sybase	404
Référence des options Teradata	409
Référence des options des fichiers structurés sous forme de tables.	413
Référence des options des services Web	416
Référence des options XML.	424

Chapitre 37. Vues de la table du catalogue global contenant des informations fédérées. 431

Chapitre 38. Options de mappage de fonctions pour les systèmes fédérés . 435

Chapitre 39. Types de serveur valides dans les instructions SQL 437

Chapitre 40. Mappages de type de données 439

Mappages de types de données aval par défaut	439
Mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows	439
Mappages de types de données aval par défaut pour les sources de données DB2 for System i	440
Mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE	441
Mappages de types de données aval par défaut pour les sources de données DB2 for z/OS	442
Mappages de types de données aval par défaut pour les sources de données Informix	442

Mappages de types de données aval par défaut pour les sources de données JDBC	444	Mappages de types de données par défaut Unicode	462
Mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server	445	Mappages de types de données aval par défaut Unicode pour les sources de données JDBC	462
Mappages de types de données aval par défaut pour les sources de données ODBC	447	Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC	462
Mappages de types de données aval par défaut pour les sources de données Oracle NET8	448	Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server	463
Mappages de types de données aval par défaut pour les sources de données Sybase	448	Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server	463
Mappages de types de données aval par défaut pour les sources de données Teradata	450	Mappages de types de données aval par défaut Unicode pour les sources de données NET8	464
Echantillons de mappages de types de données aval	451	Mappages de types de données inversés par défaut Unicode pour les sources de données NET8	464
Mappages de types de données inversés par défaut	453	Mappages de types de données aval par défaut Unicode pour les sources de données ODBC	464
Mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows	454	Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC	465
Mappages de types de données inversés par défaut pour les sources de données DB2 for System i	455	Mappages de types de données aval par défaut Unicode pour les sources de données Sybase	465
Mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE	455	Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase	466
Mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS	456	Types de données pris en charge pour les sources de données non relationnelles	466
Mappages de types de données inversés par défaut pour les sources de données Informix	456		
Mappages de types de données inversés par défaut pour les sources de données JDBC	457	Documentation du produit	471
Mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server	458	Lecture des diagrammes de syntaxe	473
Mappages de types de données inversés par défaut pour les sources de données ODBC	459	Documentation accessible	475
Mappages de types de données inversés par défaut pour les sources de données Oracle NET8	459	Remarques	477
Mappages de types de données inversés par défaut pour les sources de données Sybase	460	Marques	479
Mappages de types de données inversés par défaut pour les sources de données Teradata	461	Index	481

Avis aux lecteurs canadiens

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

Illustrations

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

Terminologie

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

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

Claviers

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








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

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

Nomenclature

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

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

Brevets

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

Assistance téléphonique

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

Chapitre 1. Systèmes fédérés

Un *système fédéré* est un type particulier de système de gestion de base de données réparties (DBMS). Un système fédéré consiste en une instance DB2 qui fonctionne comme un serveur fédéré, une base de données qui agit comme base de données fédérée, une ou plusieurs sources de données et des clients (utilisateurs et applications) qui accèdent à la base de données et aux sources de données.

Avec un système fédéré, vous pouvez envoyer des requêtes réparties à plusieurs sources de données dans une instruction SQL unique. Par exemple, vous pouvez joindre des données qui se trouvent dans une table DB2, une table Oracle et un fichier balisé XML dans une instruction SQL. La figure suivante illustre les composants d'un système fédéré et un exemple des sources de données auxquelles vous pouvez accéder.

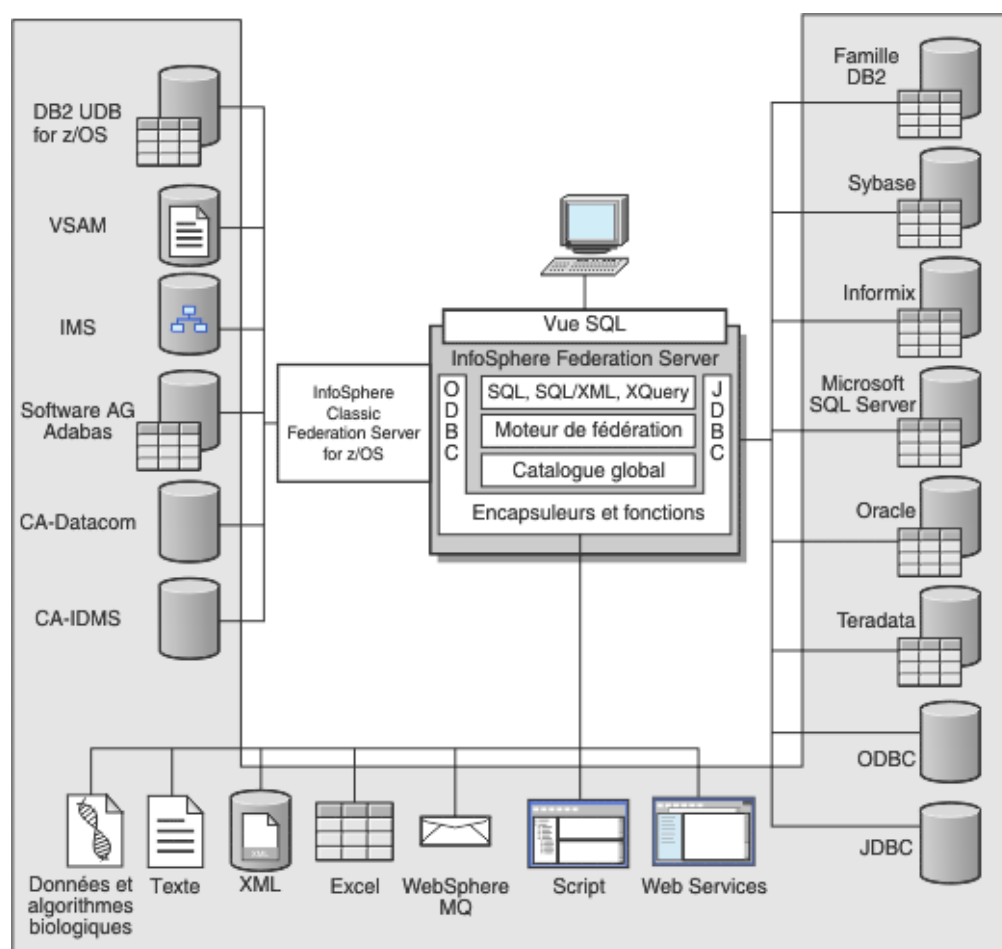


Figure 1. Composants d'un système fédéré

La puissance d'un système fédéré réside dans sa capacité à :

- Corréler des données à partir de tables locales et de sources de données distantes, comme si toutes les données étaient stockées dans la base de données fédérée

- Mettre à jour les données dans des sources de données relationnelles, comme si les données étaient stockées dans la base de données fédérée
- Déplacer les données vers et à partir des sources de données
- Tirer profit de la puissance de traitement de la source de données, en envoyant des requêtes aux sources de données pour traitement
- Compenser les limites du langage SQL au niveau de la source de données en traitant des parties d'une requête répartie au niveau du serveur fédéré

Sources de données prises en charge

Vous pouvez accéder à de nombreuses sources de données à l'aide d'un système fédéré.

IBM® InfoSphere Federation Server prend en charge les sources de données répertoriées dans les tableaux suivants. Le premier tableau répertorie les configurations requises pour le logiciel client de données. Le logiciel client doit être acquis séparément sauf indication contraire.

Vous devez installer le logiciel client pour les sources de données auxquelles vous voulez accéder. Le logiciel client doit être installé sur le même système que IBM InfoSphere Federation Server. Vous devez également disposer du SDK Java™ approprié pour utiliser certains outils comme DB2 Control Center ainsi que pour créer et exécuter des applications Java, notamment des procédures mémorisées et des fonctions définies par l'utilisateur.

Pour obtenir les informations les plus récentes, consultez la Page des configurations requises pour les sources de données sur le Web.

Tableau 1. Sources de données prises en charge, configurations logicielles client requises et prise en charge à partir des systèmes d'exploitation 32 bits.

Source de données	Versions prises en charge	Logiciel client	Système d'exploitation et architecture matérielle 32 bits	
			X86-32	X86-32
			Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
BioRS	5.2, 5.3	Aucune	Y	Y
DB2 for Linux, UNIX® et Windows	8.1.x, 8.2.x, 9.1, 9.5, 9.7	Aucune	Y	Y
DB2 for z/OS	7.x, 8.x, 9.x	DB2 Connect V9.7	Y	Y
DB2 for System i	5.2, 5.3, 5.4, 6.1	DB2 Connect V9.7	Y	Y
DB2 Server for VSE et VM	7.2 , 7.4	DB2 Connect V9.7	Y	Y
Fichiers à plat		Aucune	Y	Y
Informix	Informix XPS 8.50, 8.51 et Informix IDS 7.31, IDS 9.40, IDS 10.0, 11.5, 11.10	Informix Client SDK version 2.81.TC2 ou supérieure ; 3.0 requise pour SLES 10 sous Power	Y	Y
JDBC	3.0 ou version supérieure	Pilotes JDBC compatibles avec JDBC 3.0 ou version supérieure	Y	Y

Tableau 1. Sources de données prises en charge, configurations logicielles client requises et prise en charge à partir des systèmes d'exploitation 32 bits. (suite)

			Système d'exploitation et architecture matérielle 32 bits	
			X86-32	X86-32
Source de données	Versions prises en charge	Logiciel client	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
Microsoft® Excel	2000, 2002, 2003, 2007	Aucune		Y
Microsoft SQL Server	Microsoft SQL Server 2000/SP4, 2005, 2008	Pour Windows, pilote Microsoft SQL Server Client ODBC 3.0 (ou version supérieure). Pour Unix, DataDirect ODBC 5.3	Y	Y
MQ	MQ7	MQ7	Y	Y
ODBC	3.0	Pilotes ODBC compatibles avec ODBC 3.0, ou version supérieure**	Y	Y
OLE DB	2.7, 2.8	OLE DB 2.0 ou version supérieure	Y	Y
Oracle	10g, 10gR2, 11g, 11gR1	Client Oracle NET 10.0 - 10.1, 10.2.0.1 avec module de correction 3807408, 10.1.0.3 avec module de correction 3807408, 11.1.0.6.0	Y	Y
Sybase	Sybase ASE 12.5, 15.0	Sybase Open Client 12.5 - 15.0	Y	Y
Teradata	2.5, 2.6, 12	Composants communs partagés pour l'internationalisation de Teradata (tdicu) version 1.01 ou supérieure, Teradata Generic Security Services (TeraGSS) version 6.01 ou supérieure et le logiciel sur les systèmes d'exploitation suivants : Pour Windows client Teradata TTU 8.0 ou version supérieure ou bibliothèque d'API Teradata API CLiv2 4.8.0 ou version supérieure Pour UNIX et Linux Teradata Call-Level Interface Version 2 CLiv2 Edition 4.8.0 ou supérieure	Y	Y
Web Services	WSDL 1.0, 1.1 SOAP 1.0, 1.1	Aucune	Y	Y
XML	XML1.0, XML1.1	Aucune	Y	Y

Tableau 1. Sources de données prises en charge, configurations logicielles client requises et prise en charge à partir des systèmes d'exploitation 32 bits. (suite)

			Système d'exploitation et architecture matérielle 32 bits	
			X86-32	X86-32
Source de données	Versions prises en charge	Logiciel client	Linux [®] , RedHat Enterprise Linux (RHEL), SUSE	Windows [®]

** ODBC peut être utilisé pour accéder à RedBrick 6.20cu5 et InfoSphere Classic Federation V8.2 et versions supérieures, parmi d'autres sources de données.

Tableau 2. Prise en charge à partir des systèmes d'exploitation 64 bits.

architecture matérielle 64 bits	X86-64	X86-64	Power	Itanium [®]	Power	Sparc	zSeries
Système d'exploitation	Linux RHEL SUSE	Windows	AIX	HP-UX	Linux RHEL SUSE	Solaris	Linux RHEL SUSE
Source de données							
BioRS	Y	Y	Y	Y	Y	Y	Y
DB2 for Linux, UNIX et Windows	Y	Y	Y	Y	Y	Y	Y
DB2 for z/OS	Y	Y	Y	Y	Y	Y	Y
DB2 for System i	Y	Y	Y	Y	Y	Y	Y
DB2 Server for VSE et VM	Y	Y	Y	Y	Y	Y	Y
Informix	Y		Y	Y	Y	Y	Y
JDBC	Y	Y	Y	Y	Y	Y	Y
Microsoft Excel							
Microsoft SQL Server	Y	Y	Y	Y		Y	
MQ	Y	N	Y	Y	Y	Y	Y
ODBC	Y	Y	Y***	Y		Y***	Y
OLE DB		Y		Y			
Oracle	Y	Y	Y	Y	Y	Y	Y
Script	Y	Y	Y	Y	Y	Y	Y
Sybase	Y		Y	Y	Y	Y	
Teradata	Y		Y	Y		Y	
Web Services	Y	Y	Y	Y	Y	Y	Y
XML	Y	Y	Y	Y	Y	Y	Y

*** ODBC peut être utilisé pour accéder à RedBrick 6.20cu5 et IBM InfoSphere Classic Federation Server for z/OS à l'aide des clients 32 et 64 bits.

Le serveur fédéré

Dans un système fédéré, le serveur DB2 est qualifié de serveur fédéré. Un nombre indéfini d'instances DB2 peuvent être configurées pour fonctionner comme des serveurs fédérés. Vous pouvez utiliser des instances DB2 existantes comme serveurs fédérés, ou bien en créer d'autres spécifiques pour le système fédéré.

L'instance DB2 gérant le système fédéré est appelée serveur car elle répond aux demandes des utilisateurs finaux et des applications client. Le serveur fédéré envoie souvent des parties des requêtes qu'il reçoit aux sources de données pour traitement. Une opération de transfert de l'exécution sur la base de données est une opération qui est traitée à distance. L'instance DB2 gérant le système fédéré est qualifiée de serveur fédéré, même si elle fonctionne comme client lorsqu'elle transmet des demandes aux sources de données.

Comme tout autre serveur d'applications, le serveur fédéré est une instance du gestionnaire de bases de données. Les processus applicatifs connectent et soumettent les requêtes à la base de données au sein du serveur fédéré. Cependant, deux caractéristiques principales le distinguent des autres serveurs d'applications :

- Un serveur fédéré est configuré pour recevoir des requêtes susceptibles d'être partiellement ou entièrement destinées à des sources de données. Le serveur fédéré répartit ces requêtes vers les sources de données.
- Comme d'autres serveurs d'applications, un serveur fédéré utilise les protocoles de communication DRDA (sur TCP/IP) pour communiquer avec les instances de la famille DB2. Cependant, contrairement à d'autres serveurs d'applications, un serveur fédéré utilise le client natif de la source de données pour accéder à la source de données. Par exemple, un serveur fédéré utilise Sybase Open Client pour accéder aux sources de données Sybase et un pilote ODBC Microsoft SQL Server pour accéder aux sources de données Microsoft SQL Server.

Qu'est-ce qu'une source de données ?

Dans un système fédéré, une *source de données* peut être une base de données relationnelle (telle que Oracle ou Sybase) ou une source de données non relationnelle (telle qu'un fichier balisé XML).

Par l'intermédiaire de certaines sources de données, vous pouvez accéder à d'autres sources de données. L'encapsuleur de connectivité ODBC vous permet, par exemple, d'accéder à des IBM InfoSphere Classic Federation Server for z/OS sources de données, telles que DB2 pour z/OS, IMS, CA-IDMS, CA-Datacom, Software AG Adabas et VSAM.

La méthode ou le protocole utilisé pour accéder à une source de données dépend du type de source de données. Par exemple, l'architecture de base de données relationnelle répartie sert à accéder à des sources de données DB2 for z/OS.

Les sources de données sont autonomes. Par exemple, le serveur fédéré peut envoyer des requêtes à des sources de données Oracle au même moment où les applications Oracle accèdent à ces sources de données. Un système fédéré ne monopolise pas ou ne restreint pas l'accès à d'autres sources de données, au delà des contraintes d'intégrité et de verrouillage.

La base de données fédérée

Les sources de données apparaissent sous la forme d'une base de données collective unique dans le système de base de données DB2 pour les applications client et les utilisateurs finaux. Les utilisateurs et les applications s'interfaçent avec la *base de données fédérée* qui est gérée par le serveur fédéré.

La base de données fédérée contient un catalogue système qui stocke des informations sur les données. Le catalogue système de la base de données fédérée contient des entrées qui identifient les sources de données et leurs caractéristiques. Le serveur fédéré consulte les informations stockées dans le catalogue système de la base de données fédérée et l'encapsuleur de source de données afin de déterminer le meilleur plan pour traiter les instructions SQL.

Le système fédéré traite les instructions SQL comme si les données provenant de sources de données étaient des tables relationnelles ordinaires ou des vues au sein de la base de données fédérée. En conséquence :

- Le système fédéré peut corrélérer des données relationnelles avec des données dans des formats non relationnels. Cela est vrai lorsque les sources de données utilisent différents dialectes SQL ou ne prennent pas du tout en charge le langage SQL.
- Les caractéristiques de la base de données fédérée sont prioritaires lorsqu'il existe des différences entre les caractéristiques de la base de données fédérée et les caractéristiques des sources de données. Les résultats de la requête sont conformes à la sémantique DB2, même si des données provenant de sources de données non DB2 sont utilisées pour calculer ces résultats.

Exemples :

- La page de codes que le serveur fédéré utilise est différente de la page de codes utilisée par la source de données. Dans ce cas, les données de type caractères provenant de la source de données sont converties en fonction de la page de codes utilisée par la base de données fédérée, lorsque ces données sont renvoyées vers un utilisateur fédéré.
- La séquence de classement que le serveur fédéré utilise est différente de celle utilisée par la source de données. Dans ce cas, toute opération de tri sur des données de type caractères est effectuée au niveau du serveur fédéré à la place de la source de données.

Encapsuleurs et modules d'encapsuleur

Les *encapsuleurs* sont des mécanismes qui permettent à la base de données fédérée d'interagir avec des sources de données. La base de données fédérée utilise des routines stockées dans une bibliothèque appelée *module d'encapsuleur* pour implémenter un encapsuleur.

Ces routines permettent à la base de données fédérée d'effectuer des opérations comme la connexion à une source de données et la récupération de données à partir de cette source de manière itérative. Généralement, le propriétaire d'instance fédérée utilise l'instruction CREATE WRAPPER pour enregistrer un encapsuleur dans la base de données fédérée. Vous pouvez enregistrer un encapsuleur comme étant isolé ou sécurisé en utilisant l'option d'encapsuleur DB2_FENCED.

Vous créez un encapsuleur pour chaque type de source de données auquel vous voulez accéder. Vous souhaitez par exemple accéder à trois tables de base de données DB2 for z/OS, à une table DB2 for System i, à deux tables Informix et à une vue Informix. Dans ce cas, vous devez créer un encapsuleur pour les objets de

source de données DB2 et un autre pour les objets de source de données Informix. Une fois ces encapsuleurs enregistrés dans la base de données fédérée, vous pouvez les utiliser pour accéder à d'autres objets à partir de ces sources de données. Vous pouvez par exemple utiliser l'encapsuleur d'architecture de base de données relationnelle répartie avec tous les objets de sources de données de la famille DB2 : DB2 Database for Linux, UNIX et Windows, DB2 for z/OS, DB2 for System i et DB2 Server for VM and VSE.

Vous utilisez les définitions de serveur et les pseudonymes pour identifier les spécificités (nom, emplacement, etc.) de chaque objet de source de données.

Un encapsuleur effectue de nombreuses tâches. Il effectue par exemple les tâches suivantes :

- Il effectue la connexion à la source de données. Il utilise l'API de connexion standard de la source de données.
- Il soumet des requêtes à la source de données.
 - Pour les sources de données qui prennent en charge le langage SQL, la requête est soumise en SQL.
 - Pour les sources de données qui ne prennent pas en charge le langage SQL, la requête est convertie en langage de requête natif de la source de données ou en une série d'appels d'API source.
- Il reçoit les ensembles de résultats provenant de la source de données. L'encapsuleur utilise les API standard source pour la réception de l'ensemble de résultats.
- Il répond aux requêtes de la base de données fédérée sur les mappages de type de données par défaut pour une source de données. L'encapsuleur contient les mappages de type par défaut qui sont utilisés lorsque des pseudonymes sont créés pour un objet de source de données. Pour les encapsuleurs relationnels, les mappages de type de données que vous créez écrasent les mappages de type de données par défaut. Les mappages de type de données définis par l'utilisateur sont stockés dans le catalogue global.
- Il répond aux requêtes de la base de données fédérée sur les mappages de fonctions par défaut pour une source de données. La base de données fédérée a besoin d'informations sur le mappage de type de données à des fins de planification de la requête. L'encapsuleur contient des informations dont la base de données fédérée a besoin pour déterminer si et comment les fonctions DB2 sont mappées vers des fonctions de la source de données. Ces informations sont utilisées par le compilateur SQL pour déterminer si la source de données est capable d'effectuer les opérations de requête. Pour les encapsuleurs relationnels, les mappages de fonctions que vous créez écrasent les mappages de type de fonction par défaut. Les mappages de fonctions définis par l'utilisateur sont stockés dans le catalogue global.

Les *options d'encapsuleur* sont utilisées pour configurer l'encapsuleur ou pour définir comment IBM InfoSphere Federation Server utilise l'encapsuleur.

Comment interagir avec un système fédéré

Sachant que la base de données fédérée appartient à la famille DB2, vous pouvez interagir avec de plusieurs façons.

Vous pouvez interagir avec un système fédéré en utilisant l'une des méthodes suivantes :

- l'interpréteur de commandes (CLP) de DB2,

- L'interface graphique du centre de contrôle DB2,
- Programmes d'application
- Outils de la famille DB2
- Fournisseurs de services Web

La procédure décrite dans la documentation fédérée indique les commandes et les instructions SQL que vous pouvez entrer dans l'interpréteur de commande DB2. La documentation précise les tâches possibles via l'interface graphique du Centre de contrôle DB2. Puisque l'interface graphique du centre de contrôle DB2 est intuitive, la procédure permettant d'effectuer ces tâches via le Centre de contrôle DB2 ne figure pas dans cette documentation.

Centre de contrôle DB2

L'interface graphique du centre de contrôle DB2 vous permet d'effectuer la plupart des tâches nécessaires pour définir, configurer et modifier le système fédéré. Le centre de contrôle DB2 utilise des panneaux (boîtes de dialogue et assistants) pour vous guider dans l'exécution des tâches.

Les panneaux du centre de contrôle DB2 incluent une aide interactive lorsque votre souris survole une commande, telle qu'une zone de liste ou un bouton de commande. En outre, chaque panneau dispose d'un bouton d'aide qui fournit des informations sur la tâche du panneau et propose des liens vers des concepts connexes et des informations de référence.

Vous pouvez utiliser un assistant pour créer les objets fédérés ou créer chaque objet individuellement.

Utilisez le centre de contrôle DB2 pour configurer l'accès aux services Web et aux sources de données XML. Les fonctions intégrées au centre de contrôle DB2 simplifient les étapes à suivre pour configurer le serveur fédéré afin d'accéder à ces sources de données.

L'interface graphique du centre de contrôle DB2 est le moyen le plus facile d'effectuer les tâches de configuration de base de sources de données :

- Créer les encapsuleurs et définir les options de l'encapsuleur
- Spécifier les variables d'environnement pour votre source de données
- Créer les définitions de serveur et définir les options de serveur
- Créer les mappages utilisateur et définir les options utilisateur
- Créer les pseudonymes et définir les options de pseudonyme ou les options de colonne

Après avoir configuré le serveur fédéré pour accéder à vos sources de données, vous pouvez utiliser le centre de contrôle DB2 pour :

- Modifier la configuration de la source de données
- Surveiller le statut des pseudonymes et des serveurs
- Gérer les statistiques en cours de vos pseudonymes
- Créer et modifier des tables de mémoire cache
- Spécifier des contraintes informationnelles sur les pseudonymes
- Créer des tables distantes via IBM InfoSphere Federation Server en utilisant un langage de définition de données transparent

Programmes d'application

Les applications ne nécessitent pas de codage spécial pour fonctionner avec les données fédérées. Les applications accèdent au système exactement de la même manière que tout autre application client DB2. Les applications s'interfaçent avec la base de données fédérée qui est au sein du serveur fédéré.

Pour obtenir des données des sources de données, les applications soumettent des requêtes en langage DB2 à la base de données fédérée. La base de données fédérée répartit ensuite les requêtes vers les sources de données appropriées, collecte les données demandées et renvoie ces données aux applications. Cependant, dans la mesure où la base de données fédérée interagit avec les sources de données via des pseudonymes, vous devez avoir conscience des points suivants :

- Les restrictions SQL qui existent lorsque vous travaillez avec des pseudonymes
- Comment effectuer des opérations sur des objets mis en pseudonyme

Outils de la famille DB2

Vous pouvez interagir avec une base de données fédérée en utilisant des outils d'hôte et de milieu de gamme.

Entre autres outils d'hôte et de milieu de gamme, vous pouvez utiliser ceux qui suivent pour interagir avec une base de données fédérée :

- des outils DB2 qui améliorent les performances et la gestion de DB2 for z/OS et DB2 for Linux, UNIX et Windows,
- SQL interactif (STRSQL) sur DB2 for System i.

IBM Optim Data Studio

IBM Optim Data Studio est une solution complète de gestion de données que vous pouvez utiliser en association avec les informations auxquelles vous accédez à l'aide des fonctions fédérées de IBM InfoSphere Federation Server.

Vous pouvez utiliser les outils de Optim Data Studio pour la configuration, la conception et le développement.

- Pour la configuration : IBM Optim Data Administrator for DB2 Linux, UNIX et Windows

Optim Database Administrator simplifie la collaboration entre les administrateurs de bases de données, les architectes et les développeurs, ainsi que les processus d'identification, d'analyse et d'implémentation des modifications du schéma de base de données pour DB2 for Linux, UNIX et Windows.

- Pour la conception : IBM InfoSphere Data Architect

InfoSphere Data Architect est un environnement de développement détaillé conçu pour la modélisation des données, l'association et l'intégration des ressources de données, et le développement d'applications de base de données. Grâce à InfoSphere Data Architect, vous pouvez rechercher et mapper des relations parmi une multitude de sources de données, élaborer des scripts illustrant ces relations, puis déployer des scripts vers des serveurs locaux ou distants, fédérés ou non fédérés.

- Pour le développement : IBM Optim Development Studio

Optim Development Studio propose un environnement de développement et de test complet pour concevoir des objets de base de données, des requêtes, une logique de base de données et des applications pureQuery.

Pour plus d'informations sur Optim Data Studio, consultez le Centre de documentation sur la gestion intégrée des données, accessible à l'adresse <http://publib.boulder.ibm.com/infocenter/idm/v2r2/index.jsp>.

Fournisseurs de services Web

Vous pouvez interagir avec une base de données fédérée via des fournisseurs de services Web.

Au sein des systèmes fédérés, un encapsuleur de services Web est disponible pour accéder aux services Web avec des instructions SQL sur des pseudonymes et des vues qui appellent des services Web. Vous pouvez créer un encapsuleur de services Web et des pseudonymes qui spécifient l'entrée vers le service Web et l'accès à la sortie à partir du service Web avec les instructions SELECT.

Catalogue système des bases de données fédérées

Le catalogue système de base de données fédérée contient des informations sur les objets de la base de données fédérée et sur les objets des sources de données.

Le catalogue d'une base de données fédérée est appelée catalogue global car il contient des informations sur l'ensemble du système fédéré. L'optimiseur de requêtes DB2 se sert des informations dans le catalogue global et l'encapsuleur de source de données pour planifier la meilleure façon de traiter des instructions SQL. Les informations stockées dans le catalogue global comprennent des informations locales et distantes, comme les noms de colonne, les types de données de colonne, les valeurs par défaut de colonne, les informations d'index et les informations de statistique.

Les informations du catalogue distant sont les informations ou le nom utilisé par la source de données. Les informations du catalogue local sont les informations ou le nom utilisé par la base de données fédérée. Par exemple, supposons une table distante comprenant une colonne avec le nom *EMPNO*. Le catalogue global stockerait le nom de la colonne distante en tant que *EMPNO*. A moins que vous ne désigniez un nom différent, le nom de colonne locale sera stocké en tant que *EMPNO*. Vous pouvez modifier le nom de colonne locale en *Employee_Number*. Les utilisateurs qui soumettent des requêtes incluant cette colonne utiliseront *Employee_Number* dans leurs requêtes à la place de *EMPNO*. Vous utilisez l'instruction ALTER NICKNAME pour modifier le nom local des colonnes de source de données.

Pour les sources de données relationnelles et non relationnelles, les informations stockées dans le catalogue global comprennent à la fois des informations locales et distantes.

Pour afficher les informations de la table de source de données stockées dans le catalogue global, effectuez une requête sur les vues du catalogue SYSCAT.TABLES, SYSCAT.NICKNAMES, SYSCAT.TABOPTIONS, SYSCAT.INDEXES, SYSCAT.INDEXOPTIONS, SYSCAT.COLUMNS et SYSCAT.COLOPTIONS dans la base de données fédérée.

Le catalogue global comprend également d'autres informations sur les sources de données. Par exemple, le catalogue global comprend des informations que le serveur fédéré utilise pour se connecter à la source de données et mapper les autorisations utilisateur fédérées aux autorisations utilisateur de la source de

données. Le catalogue global contient des attributs concernant la source de données que vous définissez explicitement, comme les options de serveur.

Le compilateur SQL

Le compilateur SQL DB2 regroupe des informations utiles au traitement des requêtes.

Pour obtenir des données à partir de sources de données, les utilisateurs et les applications soumettent des requêtes en langage SQL à la base de données fédérée. Lorsqu'une requête est soumise, le compilateur SQL DB2 consulte les informations qui se trouvent dans le catalogue global et l'encapsuleur de source de données pour le traitement de cette requête. Cela comprend des informations de connexion à la source de données, des informations sur le serveur, des mappages, des informations d'indexation et des statistiques de traitement.

L'optimiseur de requêtes

Dans le cadre du processus du compilateur SQL, l'optimiseur de requêtes analyse une requête. Le compilateur développe des stratégies alternatives appelées plans d'accès pour le traitement de la requête.

Les plans d'accès peuvent demander à la requête d'être :

- Traitement par les sources de données
- Traitement par le serveur fédéré
- Traitement en partie par les sources de données et en partie par le serveur fédéré

L'optimiseur de requêtes évalue les plans d'accès essentiellement sur la base des informations relatives aux données et aux fonctionnalités de la source de données. Ces informations se trouvent dans l'encapsuleur et le catalogue global.

L'optimiseur de requêtes décompose la requête en segments appelés fragments de requête. En règle générale, il est plus efficace de transférer l'exécution d'un fragment de requêtes sur une source de données, si la source de données peut traiter le fragment. Toutefois, l'optimiseur de requêtes prend en compte d'autres facteurs tels que :

- La quantité de données à traiter
- La vitesse de traitement de la source de données
- La quantité de données que le fragment renverra
- La bande passante de communication
- Le fait qu'il existe une table de requêtes matérialisée utilisable sur le serveur fédéré qui représente le même résultat de requête

L'optimiseur de requêtes génère des alternatives de plan d'accès pour le traitement d'un fragment de requête. Les alternatives du plan effectuent différentes quantités de travail en local sur le serveur fédéré et sur les sources de données distantes.

Dans la mesure où l'optimiseur de requêtes est basé sur les coûts, il affecte les coûts de consommation des ressources aux alternatives du plan d'accès.

L'optimiseur de requêtes choisit ensuite le plan qui traitera la requête avec le coût de consommation de ressources le moins élevé.

Si l'un des fragments doit être traité par des sources de données, la base de données fédérée soumet ces fragments aux sources de données. Une fois les fragments traités par les sources de données, les résultats sont récupérés et renvoyés vers la base de données fédérée. Si la base de données fédérée effectue

une partie du traitement, elle associe ses résultats à ceux extraits de la source de données. La base de données fédérée renvoie ensuite tous les résultats au client.

Compensation

L'aptitude de IBM InfoSphere Federation Server à traiter le langage SQL non pris en charge par une source de données est appelée *compensation*.

La base de données fédérée ne transfère pas l'exécution d'un fragment de requête sur la base de données si la source de données ne peut pas le traiter ou si le serveur fédéré peut le traiter plus rapidement que la source de données. Par exemple, en supposant que le dialecte SQL d'une source de données ne prenne pas en charge un regroupement CUBE dans la clause GROUP BY. Une requête contenant un regroupement CUBE et référençant une table dans la source de données est soumise au serveur fédéré. La base de données fédérée ne transfère pas l'exécution du regroupement CUBE vers la source de données mais traite le CUBE lui-même.

La base de données fédérée compense le manque de fonctionnalités au niveau de la source de données de deux manières :

- Elle peut demander à la source de données d'utiliser une ou plusieurs opérations équivalentes à la fonction DB2 indiquée dans la requête. Par exemple avec une source de données qui ne prend pas en charge la fonction cotangente (COT(x)) mais qui prend en charge la fonction tangente (TAN(x)). La base de données fédérée peut demander à la source de données d'effectuer le calcul (1/TAN(x)), qui est l'équivalent de la fonction cotangente (COT(x)).
- Elle peut renvoyer l'ensemble des données vers le serveur fédéré et effectuer la fonction localement.

Pour les sources de données relationnelles, chaque type de système de gestion de base de données relationnelle prend en charge un sous-ensemble du standard SQL international. En outre, certains types de système de gestion de base de données relationnelle prennent en charge des constructions SQL qui vont au-delà de ce standard. Un *dialecte SQL* correspond à la totalité du langage SQL qu'un type de SGBD relationnel prend en charge. Si une construction SQL se trouve dans le dialecte SQL DB2 mais pas dans celui de la source de données relationnelle, le serveur fédéré peut implémenter cette construction au nom de la source de données.

La base de données fédérée peut compenser les différences qui existent entre les dialectes SQL. La clause d'expression de table commune en est un exemple. Le langage DB2 SQL comprend la clause common-table-expression. Dans cette clause, un nom peut être spécifié de telle manière que toutes les clauses FROM d'une instruction FULLSELECT puissent référencer un ensemble de résultats. Le serveur fédéré traitera une expression de table commune pour une source de données même si le dialecte SQL utilisé par la source de données ne comprend pas l'expression de table commune.

Grâce à la compensation, la base de données fédérée peut prendre en charge l'ensemble du dialecte SQL DB2 pour les requêtes de sources de données. Même les sources de données disposant d'une prise en charge limitée ou inexistante du langage SQL bénéficieront de la compensation. Vous devez utiliser le dialecte SQL DB2 avec un système fédéré, sauf dans une session passe-système.

Sessions passe-système

Vous pouvez soumettre des instructions SQL directement aux sources de données à l'aide d'un mode spécial appelé *passe-système*.

Vous soumettez des instructions SQL dans le dialecte SQL utilisé par la source de données. Utilisez une session passe-système pour exécuter une opération qui n'est pas réalisable à l'aide de DB2 SQL/API. Par exemple, utilisez une session passe-système pour créer une procédure ou un index ou effectuez des requêtes dans le dialecte natif de la source de données.

Actuellement, les sources de données qui prennent en charge le passe-système utilisent le langage SQL. A l'avenir, il est possible que les sources de données prendront en charge le passe-système en utilisant un langage de source de données différent du langage SQL.

De la même manière, vous pouvez utiliser une session passe-système pour effectuer des actions qui ne sont pas prises en charge par SQL, comme certaines tâches d'administration. Cependant, vous ne pouvez pas utiliser de session passe-système pour effectuer toutes les tâches administratives. Par exemple, vous pouvez créer ou supprimer des tables au niveau de la source de données, mais vous ne pouvez pas démarrer ou arrêter la base de données distante.

Vous pouvez utiliser le langage SQL statique et dynamique dans une session passe-système.

Le serveur fédéré fournit les instructions SQL suivantes pour gérer les sessions passe-système :

SET PASSTHRU

Ouvre une session passe-système. Lorsque vous émettez une autre instruction SET PASSTHRU pour démarrer une nouvelle session passe-système, la session passe-système en cours est fermée.

SET PASSTHRU RESET

Ferme la session passe-système en cours.

GRANT (privilège serveur)

Accorde à un utilisateur, un groupe, une liste d'ID d'autorisation ou à PUBLIC des droits d'accès permettant d'initier des sessions passe-système vers une source de données spécifique.

REVOKE (privilège serveur)

Révoque les droits d'accès permettant d'initier des sessions passe-système.

Définitions et options de serveur

Une fois les encapsuleurs créés pour les sources de données, le propriétaire d'instance fédérée définit les sources de données vers la base de données fédérée.

Le propriétaire d'instance fournit un nom permettant d'identifier la source de données et d'autres informations qui se rapportent à la source de données. Ces informations comprennent :

- Le type et la version de la source de données
- Le nom de la base de données pour la source de données (SGBD relationnel uniquement)
- Des métadonnées spécifiques à la source de données

Par exemple, une source de données de la famille DB2 peut avoir plusieurs bases de données. La définition doit spécifier la base de données à laquelle le serveur fédéré peut se connecter. En revanche, une source de données Oracle n'a qu'une seule base de données et le serveur fédéré peut s'y connecter sans connaître son nom. Le nom de la base de données n'est pas incluse dans la définition du serveur fédéré d'une source de données Oracle.

Le nom et les autres informations fournis par le propriétaire d'instance au serveur fédéré sont collectivement appelés *définition de serveur*. Les sources de données répondent à des requêtes concernant des données et constituent elles-mêmes des serveurs.

Les instructions CREATE SERVER et ALTER SERVER sont utilisées pour créer et modifier une définition de serveur.

Certaines des informations d'une définition de serveur sont stockées sous forme d'*options de serveur*. Lorsque vous créez des définitions de serveur, il est important de comprendre les options que vous pouvez spécifier sur le serveur.

Les options de serveur peuvent être définies de manière à être conservées pour plusieurs connexions à la source de données ou pour la durée d'une seule connexion.

Mappages utilisateur

Un *mappage utilisateur* est une association entre un identificateur utilisateur sur le serveur fédéré et les informations obligatoires pour se connecter à la source de données éloignée.

Pour créer un mappage utilisateur, vous employez l'instruction CREATE USER MAPPING. Dans cette instruction, vous indiquez l'identificateur utilisateur local, le nom local du serveur de la source de données éloignée tel que figurant dans la définition de serveur, ainsi que l'ID et le mot de passe éloignés.

Imaginez par exemple avoir créé une définition de serveur pour un serveur distant et entré "argon" comme nom local pour ce dernier. Pour permettre à Marie d'accéder à ce serveur, créer le mappage utilisateur suivant :

```
CREATE
USER MAPPING FOR Marie
SERVER argon
OPTIONS (REMOTE_AUTHID 'ID_éloigné', REMOTE_PASSWORD 'motdepasse_éloigné')
```

Lorsque Marie émet l'Instruction SQL pour se connecter au serveur distant, le serveur fédéré effectue ce qui suit :

1. Il extrait le mappage utilisateur de Marie.
2. Il déchiffre le mot de passe éloigné 'motdepasse_éloigné' associé au serveur distant.
3. Il appelle l'encapsuleur pour se connecter au serveur distant.
4. Il transmet à l'encapsuleur l'ID éloigné 'ID_éloigné' et le mot de passe éloigné déchiffré.
5. Il établit une connexion au serveur distant pour Marie.

Par défaut, le serveur fédéré stocke le mappage utilisateur dans la vue SYSCAT.USEROPTIONS du catalogue global et chiffre les mots de passe éloignés. Vous pouvez également recourir à un référentiel externe, tel qu'un fichier ou un

serveur LDAP, pour y stocker des mappages utilisateur. Pour obtenir une interface entre le serveur fédéré et le référentiel externe, vous créez un module d'extension du mappage utilisateur.

Quel que soit le mode de stockage des mappages utilisateur, prenez soin d'en limiter l'accès. Si les mappages utilisateur sont trop exposés, les données contenues dans les bases de données distantes risquent de permettre des activités non autorisées.

Pseudonymes et objets de source de données

Un *pseudonyme* est un identificateur que vous utilisez pour identifier l'objet de source de données auquel vous voulez accéder. L'objet identifié par un pseudonyme est référencé en tant qu'*objet de source de données*.

Un pseudonyme est un nom alternatif mais ne constitue pas un nom alternatif d'un objet de source de données. Un pseudonyme constitue le pointeur permettant au serveur fédéré de référencer l'objet. Les pseudonymes sont généralement définis par l'instruction CREATE NICKNAME avec des options de pseudonymes et des options de colonne de pseudonymes spécifiques.

Lorsqu'une application client ou un utilisateur soumet une requête distribuée au serveur fédéré, la requête n'a pas besoin de spécifier les sources de données. A la place, la requête référence les objets de source de données par leur pseudonyme. Les pseudonymes sont mappés vers des objets spécifiques au niveau de la source de données. Ces mappages permettent d'éviter de qualifier les pseudonymes par des noms de source de données. L'emplacement des objets de source de données est transparent pour l'application client ou l'utilisateur.

Supposons que vous définissiez le pseudonyme *DEPT* pour représenter une table de base de données Informix appelée *NFX1.PERSON*. L'instruction SELECT * FROM *DEPT* est autorisée à partir du serveur fédéré. Cependant, l'instruction SELECT * FROM *NFX1.PERSON* n'est pas autorisée à partir du serveur fédéré (sauf dans une session passe-système) à moins qu'il existe une table locale sur le serveur fédéré appelée *NFX1.PERSON*.

Lorsque vous créez un pseudonyme pour un objet de source de données, des métadonnées concernant l'objet sont ajoutées au catalogue global. L'optimiseur de requêtes utilise ces métadonnées ainsi que les informations dans l'encapsuleur pour faciliter l'accès à l'objet de source de données. Par exemple, si un pseudonyme est destiné à une table contenant un index, le catalogue global contient des informations sur l'index et l'encapsuleur contient les mappages entre les types de données DB2 et les types de données de la source de données.

Les pseudonymes pour les objets utilisant le mécanisme de sécurité LBAC (Label-Based Access Control) ne sont pas placés dans la mémoire cache. Ainsi, les données de l'objet restent sécurisées. Par exemple, si vous utilisez l'encapsuleur Oracle (Net8) pour créer un pseudonyme sur une table utilisant Oracle Label Security, la table est automatiquement identifiée comme sécurisée. Les données de pseudonymes résultantes ne peuvent pas être placées dans la mémoire cache. En conséquence, il est impossible de créer des tables de requêtes matérialisées. L'utilisation de LBAC garantit le fait que les informations ne soient consultables que par les utilisateurs disposant des privilèges de sécurité appropriés. Pour les pseudonymes créés avant le support de LBAC, vous devez employer l'instruction ALTER NICKNAME pour désactiver la mise en cache. LBAC est pris en charge par

DRDA (pour les sources de données qui utilisent DB2 for Linux, UNIX et Windows, version 9.1 et supérieure) et l'encapsuleur Net8.

Objets de source de données valides

Les pseudonyme identifient les objets de la source de données auxquels vous voulez accéder. Le tableau suivant répertorie les types d'objets pour lesquels vous pouvez créer un pseudonyme dans un système fédéré.

Tableau 3. Objets de source de données valides

Source de données	Objets valides
BioRS	Banques de données BioRS
DB2 Database pour Linux, UNIX et Windows	Pseudonymes, tables de requêtes matérialisées, tables, vues
DB2 for System i	Tables, vues, fichiers et types de table P/L (physique/logique)
DB2 pour VM et VSE	Tables, vues
DB2 for z/OS	Tables, vues
Informix	Tables, vues, synonymes
JDBC	Tous les types de table
Microsoft Excel	Fichiers .xls (seule la première feuille du classeur est accessible)
Microsoft SQL Server	Tables, vues
ODBC	Tables, vues
Oracle	Tables, vues, synonymes
Script	Scripts
Sybase	Tables, vues
Teradata	Tables, vues
Fichiers structurés sous forme de table	Fichiers texte correspondant à un format spécifique.
Web Services	Opérations dans un fichier de langage de description de services Web
Fichiers balisés XML	Jeux d'éléments dans un document XML

Options de colonne de pseudonymes

Vous pouvez ajouter au catalogue global des informations de métadonnées supplémentaires concernant l'objet mis en pseudonyme. Ces métadonnées décrivent des valeurs dans certaines colonnes de l'objet de source de données. Vous affectez ces métadonnées à des paramètres appelés *options de colonne de pseudonyme*.

Les options de colonne de pseudonyme indiquent à l'encapsuleur de gérer les données d'une colonne de manière différente d'une gestion classique. Le compilateur SQL et l'optimiseur de requêtes utilisent les métadonnées pour développer de meilleurs plans permettant d'accéder aux données.

Les options de colonne de pseudonyme sont également utilisées pour fournir d'autres informations à l'encapsuleur. Par exemple, pour les sources de données XML, une option de colonne de pseudonyme est utilisée pour indiquer à

l'encapsuleur l'expression XPath à utiliser lorsque l'encapsuleur effectue l'analyse syntaxique de la colonne du document XML.

Grâce à la fédération, le serveur DB2 traite l'objet de source de données auquel un pseudonyme fait référence comme s'il s'agissait d'une table DB2 locale. En conséquence, vous pouvez définir des options de colonne de pseudonyme pour tout objet de source de données pour lequel vous créez un pseudonyme. Certaines options de colonne de pseudonyme sont conçues pour des types spécifiques de sources de données et ne peuvent être appliquées qu'à ces sources de données.

Supposons qu'une source de données dispose d'une séquence de classement différente de celle de la base de données fédérée. Le serveur fédéré ne trie généralement pas les colonnes contenant des données de type caractères sur la source de données. Il va renvoyer les données à la base de données fédérée et effectuer un tri localement. Cependant, supposez que la colonne soit un type de données caractères (CHAR ou VARCHAR) et contienne uniquement des caractères numériques ('0','1',..., '9'). Vous pouvez l'indiquer en affectant la valeur 'Y' à l'option de colonne de pseudonyme NUMERIC_STRING. De cette façon, l'optimiseur de requêtes DB2 peut effectuer le tri à la source de données. Si le tri est effectué à distance, vous évitez la surcharge engendrée par le portage des données sur le serveur fédéré et l'opération de tri au niveau local.

Vous pouvez définir les options de colonne de pseudonyme pour les pseudonymes relationnels en utilisant les instructions ALTER NICKNAME. Vous pouvez définir les options de colonne de pseudonyme pour les pseudonymes non relationnels en utilisant les instructions CREATE NICKNAME et ALTER NICKNAME.

Mappages de type de données

Les types de données d'une source de données doivent être mappés avec ceux DB2 correspondants afin que le serveur fédéré puisse récupérer les données à partir des sources de données.

Voici des exemples de mappages de types de données par défaut :

- Le type Oracle FLOAT mappe vers le type DB2 DOUBLE.
- Le type Oracle DATE mappe vers le type DB2 TIMESTAMP.
- Le type DB2 for z/OS DATE mappe vers le type DB2 DATE.

Pour la plupart des sources de données, les mappages de type par défaut figurent dans les encapsuleurs. Les mappages de type par défaut pour les sources de données DB2 se trouvent dans l'encapsuleur DRDA. Les mappages de type par défaut pour Informix se trouvent dans l'encapsuleur INFORMIX, et ainsi de suite.

Pour certaines sources de données non relationnelles, vous devez spécifier les informations de type de données dans l'instruction CREATE NICKNAME. Les types de données DB2 correspondants doivent être indiqués pour chaque colonne de l'objet de source de données à la création du pseudonyme. Chaque colonne doit être mappée à une zone ou colonne particulière dans l'objet de source de données.

Pour les sources de données relationnelles, vous pouvez écraser les mappages de type de données par défaut. Par exemple, le type de données Informix INTEGER mappe par défaut vers le type de données DB2 INTEGER. Vous pouvez écraser les mappages par défaut et mapper le type de données Informix INTEGER vers le type de données DB2 DECIMAL(10,0).

Mappages de fonction

Pour que le serveur fédéré reconnaisse une fonction de source de données, elle doit être mappée vers une fonction homologue existante dans DB2 Database for Linux, UNIX et Windows.

IBM InfoSphere Federation Server fournit des mappages par défaut entre des fonctions existantes de source de données et des fonctions DB2 homologues. Pour la plupart des sources de données, les mappages de fonctions par défaut figurent dans les encapsuleurs. Les mappages de fonctions par défaut vers des fonctions DB2 for z/OS se trouvent dans l'encapsuleur DRDA. Les mappages de fonctions par défaut vers les fonctions Sybase figurent dans l'encapsuleur CTLIB et ainsi de suite.

Pour les sources de données relationnelles, vous pouvez créer un mappage de fonctions lorsque vous voulez utiliser une fonction de source de données que le serveur fédéré ne reconnaît pas. Le mappage que vous créez a lieu entre la fonction de source de données et une fonction DB2 homologue au niveau de la base de données fédérée. Les mappages de fonctions sont généralement utilisés lorsqu'une nouvelle fonction intégrée ou une nouvelle fonction définie par l'utilisateur devient disponible au niveau de la source de données. Les mappages de fonctions sont également utilisés lorsqu'il n'existe pas de fonction DB2 homologue. Dans ce cas, vous devez créer un modèle de fonction.

Spécifications de l'index

Lorsque vous créez un pseudonyme pour une table de source de données, les informations concernant les index de la table de source de données sont ajoutées au catalogue global. L'optimiseur de requêtes utilise ces informations pour accélérer le traitement des requêtes distribuées. Les informations du catalogue relatives à un index de source de données constituent un ensemble de métadonnées et sont appelées spécification d'index.

L'optimiseur de requêtes utilise ces informations pour accélérer le traitement des requêtes distribuées.

Un serveur fédéré ne génère pas de spécification d'index lorsque vous créez un pseudonyme pour les objets suivants :

- Une table n'ayant pas d'index
- Une vue qui ne dispose généralement pas d'informations d'index stockées dans le catalogue distant
- Un objet de source de données n'ayant pas de catalogue distant à partir duquel le serveur fédéré peut obtenir des informations d'index

Supposons qu'une table acquiert un nouvel index, en plus de ceux qu'elle avait lorsque le pseudonyme a été créé. Puisque les informations d'index sont fournies au catalogue global au moment où le pseudonyme est créé, le serveur fédéré n'a pas connaissance du nouvel index. De même, lorsqu'un pseudonyme est créé pour une vue, le serveur fédéré n'a pas connaissance de la table sous-jacente (et de ses index) à partir de laquelle la vue a été générée. Dans ces circonstances, vous pouvez fournir au catalogue global les informations d'index nécessaires. Vous pouvez créer une spécification d'index pour les tables n'ayant pas d'index. La spécification d'index indique à l'optimiseur de requêtes la colonne ou les colonnes dans lesquelles la recherche doit être effectuée pour trouver des données rapidement.

Procédures mémorisées fédérées

L'accès à la procédure fédérée permet aux utilisateurs de systèmes fédérés d'accéder à des procédures mémorisées distantes sur des sources de données distantes.

Une procédure mémorisée fédérée est une procédure stockée localement qui est mappée vers une procédure mémorisée sur une source de données. Vous utilisez l'instruction CREATE PROCEDURE (dérivée) pour enregistrer une procédure mémorisée fédérée.

Séquences de classement

L'ordre dans lequel les données de caractère sont triées dans une base de données dépend de la structure des données et de la séquence de classement définie pour la base de données.

Supposons que les données d'une base de données soient toutes en lettres majuscules et qu'elles ne contiennent aucun caractère numérique ou spécial. Un tri des données se traduirait par le même résultat, peu importe si les données ont été triées au niveau de la source de données ou de la base de données fédérée. La séquence de classement utilisée par chaque base de données ne doit pas avoir d'impact sur les résultats du tri. De la même manière, si les données de la base de données sont toutes en lettres minuscules ou en caractères numériques, un tri des données produirait les mêmes résultats sans tenir compte de l'endroit où le tri est effectué.

Si les données comprennent l'une des structures suivantes :

- Une combinaison de lettres et de caractères numériques
- Des lettres majuscules et minuscules
- Des caractères spéciaux comme @, #, €

Le tri de ces données pourrait produire des résultats différents, si la base de données fédérée et la source de données utilisent des séquences de classement différentes.

D'un point de vue général, une séquence de classement est un classement défini de données de type caractères qui détermine si un caractère particulier effectue un tri supérieur, inférieur ou de même niveau qu'un autre caractère.

Comment les séquences de classement déterminent les ordres de tri

Une séquence de classement détermine l'ordre de tri des caractères dans un jeu de caractères codés.

Un jeu de caractères est un groupe de caractères utilisés dans un système informatique ou dans un langage de programmation. Dans un jeu de caractères codé, chaque caractère est affecté à un nombre différent sur une plage allant de 0 à 255 (ou l'équivalent hexadécimal). Les nombres sont appelés points de code ; les affectations de nombres aux caractères dans un ensemble sont collectivement appelées page de codes.

Outre le fait d'être affecté à un caractère, un point de code peut être mappé vers la position du caractère dans un ordre de tri. En termes techniques, une séquence de classement est le mappage collectif de points de code d'un jeu de caractères vers

les positions de l'ordre de tri des caractères du jeu. Une position du caractère est représentée par un nombre ; ce nombre est appelé le poids du caractère. Dans la séquence de classement la plus simple, appelée séquence d'identité, les poids sont identiques aux points de code.

Exemple : la base de données ALPHA utilise la séquence de classement par défaut de la page de codes EBCDIC. La base de données BETA utilise la séquence de classement par défaut de la page de codes ASCII. Les ordres de tri pour les chaînes de caractères dans ces deux bases de données seraient différents :

```
SELECT.....
```

```
ORDER BY COL2
```

EBCDIC-Based Sort	ASCII-Based Sort
COL2	COL2
----	----
V1G	7AB
Y2W	V1G
7AB	Y2W

Exemple : de la même façon, les comparaisons de caractères dans une base de données dépendent de la séquence de classement définie pour cette base. La base de données ALPHA utilise la séquence de classement par défaut de la page de codes EBCDIC. La base de données BETA utilise la séquence de classement par défaut de la page de codes ASCII. Les comparaisons de caractères dans ces deux bases de données donneraient des résultats distincts :

```
SELECT.....
```

```
WHERE COL2 > 'TT3'
```

EBCDIC-Based Results	ASCII-Based Results
COL2	COL2
----	----
TW4	TW4
X82	X82
39G	

Définition de la séquence de classement locale pour optimiser les requêtes

Les administrateurs peuvent créer des bases de données fédérées avec une séquence de classement particulière qui corresponde à une séquence de classement de source de données.

Puis, pour chaque définition de serveur de source de données, l'option de serveur `COLLATING_SEQUENCE` est définie sur 'Y'. Ce paramétrage indique à la base de données fédérée que les séquences de classement de la base de données fédérée et de la source de données correspondent.

Vous définissez la séquence de classement de la base de données fédérée dans le cadre de la commande `CREATE DATABASE`. Grâce à cette commande, vous pouvez indiquer l'une des séquences suivantes :

- Une séquence d'identité
- Une séquence système (la séquence utilisée par le système d'exploitation qui prend en charge la base de données)
- Une séquence personnalisée (séquence prédéfinie fournie par le système de base de données DB2 ou que vous créez)

Imaginez que la source de données est DB2 for z/OS. Les tris qui sont définis dans une clause ORDER BY sont implémentés par une séquence de classement basée sur une page de codes EBCDIC. Pour récupérer les données DB2 for z/OS triées conformément aux clauses ORDER BY, configurez la base de données fédérée de telle manière qu'elle utilise la séquence de classement prédéfinie basée sur la page de codes EBCDIC appropriée.

Chapitre 2. Modification des configurations de sources de données

A terme, vous devrez peut-être modifier les configurations de sources de données de votre système fédéré. Par exemple, si vos sources de données évoluent, il vous faudra peut-être mettre à jour les définitions de serveur, les pseudonymes, et les mappages utilisateur. Vous devrez peut-être également ajouter ou supprimer un accès aux sources de données depuis votre système fédéré.

Modification des encapsuleurs

- Modification d'un encapsuleur (Centre de contrôle DB2)
- Modification d'un encapsuleur (Ligne de commande DB2)

Modification des définitions et options de serveur

- Modification des définitions et options de serveur
- Utilisation des options de serveur dans les définitions de serveur (Centre de contrôle DB2)
- Utilisation des options de serveur dans les définitions de serveur (Ligne de commande DB2)

Modification des mappages utilisateur

- Modification d'un mappage utilisateur (Centre de contrôle DB2)
- Modification d'un mappage utilisateur (Ligne de commande DB2)

Modification des pseudonymes

- Modification d'un pseudonyme (Centre de contrôle DB2)
- Modification d'un pseudonyme (Ligne de commande DB2)

Suppression des encapsuleurs, définitions de serveur, mappages utilisateur et pseudonymes

- Suppression d'un encapsuleur
- Suppression d'une définition de serveur
- Suppression d'un mappage utilisateur
- Suppression d'un pseudonyme

Modification d'un encapsuleur (Centre de contrôle DB2)

Après avoir configuré un encapsuleur, vous pouvez utiliser l'instruction ALTER WRAPPER pour modifier la configuration en fonction de la configuration requise pour le système. Vous pouvez modifier un encapsuleur en utilisant le Centre de contrôle DB2 ou la ligne de commande DB2.

Avant de commencer

L'ID autorisation associé à l'instruction doit détenir le droit d'accès SYSADM ou DBADM.

Restrictions

Vous ne pouvez pas supprimer l'option d'encapsuleur DB2_FENCED.

Le serveur fédéré ne peut pas traiter une instruction ALTER WRAPPER dans une unité d'oeuvre donnée si cette dernière contient déjà l'une des instructions suivantes :

- une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue sur la source de données incluse dans l'encapsuleur ;
- un curseur ouvert sur un pseudonyme pour une table ou une vue sur la source de données incluse dans l'encapsuleur ;
- une insertion, suppression ou mise à jour émise pour un pseudonyme d'une table ou d'une vue sur la source de données incluse dans l'encapsuleur.

A propos de cette tâche

Cette tâche indique comment modifier un encapsuleur à partir du Centre de contrôle DB2.

Procédure

Pour modifier un encapsuleur à partir du Centre de contrôle DB2 :

1. Développez le dossier Objets de base de données fédérée. Les objets d'encapsuleur sont affichés dans le panneau du contenu de la fenêtre Centre de contrôle DB2.
2. Cliquez avec le bouton droit de la souris sur l'encapsuleur que vous souhaitez modifier puis cliquez sur **Modification** dans la liste des actions. Le bloc-notes Modification de l'encapsuleur s'ouvre.
 - a. Dans la page Paramètres, effectuez les modifications.
 - b. Cliquez sur **Définition des variables** pour définir les variables d'environnement de source de données pour l'encapsuleur. Les variables d'environnement ne sont pas requises pour tous les encapsuleurs.
3. Cliquez sur **OK** pour modifier l'encapsuleur et fermer le bloc-notes Modification de l'encapsuleur.

Modification d'un encapsuleur - exemples

Cette rubrique fournit des exemples de modification d'options d'encapsuleur à l'aide de l'instruction ALTER WRAPPER.

Pour modifier l'option DB2_FENCED en 'Y' pour l'encapsuleur appelé drda, émettez l'instruction suivante :

```
ALTER WRAPPER drda OPTIONS (SET DB2_FENCED 'Y');
```

Pour modifier l'option MODULE en '/opt/odbc/lib/libodbc.a(odbc.so)' pour l'encapsuleur appelé odbc, émettez l'instruction suivante :

```
ALTER WRAPPER odbc OPTIONS (SET MODULE '/opt/odbc/lib/libodbc.a(odbc.so)');
```

Modification d'un encapsuleur (ligne de commande DB2)

Après avoir configuré un encapsuleur, vous pouvez utiliser l'instruction ALTER WRAPPER pour modifier la configuration en fonction de la configuration requise pour le système. Vous pouvez modifier un encapsuleur en utilisant le Centre de contrôle DB2 ou la ligne de commande DB2.

Avant de commencer

L'ID autorisation associé à l'instruction doit détenir le droit d'accès SYSADM ou DBADM.

Restrictions

Vous ne pouvez pas supprimer l'option d'encapsuleur DB2_FENCED.

Le serveur fédéré ne peut pas traiter une instruction ALTER WRAPPER dans une unité d'oeuvre donnée si cette dernière contient déjà l'une des instructions suivantes :

- une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue sur la source de données incluse dans l'encapsuleur ;
- un curseur ouvert sur un pseudonyme pour une table ou une vue sur la source de données incluse dans l'encapsuleur ;
- une insertion, suppression ou mise à jour émise pour un pseudonyme d'une table ou d'une vue sur la source de données incluse dans l'encapsuleur.

A propos de cette tâche

Cette tâche indique comment modifier un encapsuleur à partir de la ligne de commande DB2. Vous pouvez utiliser l'instruction ALTER WRAPPER pour ajouter, définir ou supprimer une ou plusieurs options d'encapsuleur.

Procédure

Pour modifier un encapsuleur à partir de la ligne de commande DB2, émettez l'instruction ALTER WRAPPER.

Modification des définitions et des options de serveur

Vous utilisez l'instruction ALTER SERVER pour modifier une définition de serveur. Certaines des informations d'une définition de serveur sont stockées sous forme d'options de serveur. Lorsque vous modifiez une définition de serveur, il est important de comprendre les options que vous pouvez spécifier sur le serveur.

Une définition de serveur identifie une source de données auprès de la base de données fédérée. La définition de serveur se compose d'un nom local et d'autres informations sur ce serveur de source de données. La définition de serveur est employée par l'encapsuleur lorsque des instructions SQL utilisant des pseudonymes sont envoyées à la base de données fédérée.

Pour les sources de données relationnelles, les options de serveur peuvent être définies temporairement à l'aide de l'instruction SET SERVER OPTION. Cette instruction se substitue à la valeur de l'option de serveur dans la définition de serveur pour la durée d'une connexion unique à la base de données fédérée. Pour le SQL statique, l'utilisation de l'instruction SET SERVER OPTION affecte uniquement l'exécution de l'instruction SQL statique. L'utilisation de l'instruction SET SERVER OPTION n'a aucun effet sur les plans générés par l'optimisation.

Modifiez une définition de serveur dans les cas suivants :

- vous effectuez une mise à niveau vers une nouvelle version de la source de données ;
- vous souhaitez apporter la même modification à toutes les définitions de serveur pour un type de source de données particulier ;
- vous souhaitez ajouter ou modifier une option de serveur sur une définition de serveur existante.

Restrictions applicables à la modification de définitions de serveur

Lorsque vous modifiez des définitions de serveur, certaines restrictions s'appliquent.

Les restrictions suivantes s'appliquent à la modification de définitions de serveur :

- Dans l'instruction ALTER SERVER, vous ne pouvez pas indiquer d'encapsuleur qui ne soit pas enregistré auprès du serveur fédéré.
- Le serveur fédéré ne peut pas traiter une instruction ALTER SERVER dans une unité d'oeuvre donnée lorsque l'une des conditions suivantes se produit :
 - L'instruction fait référence à une source de données unique et l'unité d'oeuvre contient déjà l'une des instructions suivantes :
 - une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue dans la source de données ;
 - un curseur ouvert sur un pseudonyme pour une table ou une vue dans la source de données ;
 - une insertion, suppression ou mise à jour émise sur un pseudonyme d'une table ou d'une vue dans la source de données ;
 - L'instruction fait référence à une catégorie de sources de données (par exemple, toutes les sources de données d'un type et d'une version particulières) et l'unité d'oeuvre contient déjà l'une des instructions suivantes :
 - une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue dans l'une des sources de données ;
 - un curseur ouvert sur un pseudonyme pour une table ou une vue dans l'un des sources de données ;
 - une insertion, suppression ou mise à jour émise sur un pseudonyme d'une table ou d'une vue dans l'une des sources de données.
- Le serveur fédéré ne vérifie pas si la version du serveur que vous indiquez correspond à celle du serveur distant. La spécification d'un serveur incorrect peut donner lieu à des erreurs SQL lorsque vous accédez à des pseudonymes appartenant à la définition de serveur DB2. Ces erreurs ont plus de chances de se produire lorsque vous indiquez une version de serveur postérieure à la version réelle du serveur distant. Dans ce cas, lorsque vous accédez à des pseudonymes qui appartiennent à la définition de serveur, le serveur fédéré risque d'envoyer au serveur distant des instructions SQL qu'il ne reconnaît pas. Dans l'instruction ALTER SERVER, cette situation ne s'applique qu'à la forme de l'instruction qui modifie la version du serveur (*nom-serveur* VERSION version-serveur).
- Vous devez indiquer le nom complet d'option de serveur. Par exemple, vous ne pouvez pas indiquer l'abréviation DB2_TWO_PHASE. A la place, vous devez spécifier le nom complet DB2_TWO_PHASE_COMMIT.

Modification de la version de la source de données dans une définition de serveur (Centre de contrôle DB2)

Vous pouvez modifier une définition de serveur existante pour modifier la version de la source de données utilisée par le serveur distant. Vous pouvez modifier une définition de serveur à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction ALTER SERVER doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Voir Restrictions applicables à la modification de définitions de serveur.

Procédure

Pour modifier une définition de serveur à partir du Centre de contrôle DB2 :

1. Développez le dossier Objets de base de données fédérée. Les objets de définition de serveur sont affichés dans le panneau du contenu de la fenêtre Centre de contrôle DB2.
2. Cliquez avec le bouton droit de la souris sur la définition de serveur que vous souhaitez modifier puis cliquez sur **Modification** dans la liste des actions. Le bloc-notes Modification d'une définition de serveur s'ouvre.
3. Sur la page Serveur, cliquez sur la flèche **Version** pour indiquer une version différente de la source de données.
4. Cliquez sur **OK** pour modifier la définition de serveur et fermer le bloc-notes Modification d'une définition de serveur.

Modification de la version de la source de données dans une définition de serveur (ligne de commande DB2)

Vous pouvez modifier une définition de serveur existante pour modifier la version de la source de données utilisée par le serveur distant. Vous pouvez modifier une définition de serveur à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction ALTER SERVER doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Voir Restrictions applicables à la modification de définitions de serveur.

Procédure

Pour modifier une définition de serveur à partir de la ligne de commande DB2, émettez l'instruction ALTER SERVER.

Exemple : Vous utilisez une définition de serveur pour une source de données Microsoft SQL Server version 6.5. Le nom que vous avez attribué au serveur dans l'instruction CREATE SERVER est `SQLSVR_ASIA`. Si Microsoft SQL Server est mis à niveau à la version 7.0, l'instruction suivante modifie la définition du serveur :

```
ALTER SERVER SQLSVR_ASIA VERSION 7
```

Modification de toutes les définitions de serveur pour un type de source de données particulier

Vous pouvez modifier toutes les définitions de serveur existantes pour un type de source de données particulier à l'aide d'une instruction ALTER SERVER unique. Utilisez une seule instruction lorsque vous souhaitez appliquer la même modification à toutes les définitions de serveur du même type.

Avant de commencer

L'ID autorisation qui émet l'instruction ALTER SERVER doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Vous pouvez définir ou supprimer des options de serveur à l'aide de l'instruction ALTER SERVER pour un type entier de sources de données uniquement si ces options ont été ajoutées par une instruction ALTER SERVER antérieure.

Procédure

Pour modifier toutes les définitions de serveur existantes pour un type de source de données particulier, émettez une instruction ALTER SERVER unique.

Exemple : Cinq serveurs Sybase sont enregistrés dans le catalogue global de vos sources de données Sybase. Dès qu'un ID utilisateur est envoyé à l'un de ces serveurs Sybase pour authentification, vous souhaitez que le serveur fédéré convertisse systématiquement l'ID utilisateur en majuscules. En outre, vous souhaitez définir la durée pendant laquelle le serveur fédéré attend une réponse de la part de ces serveurs Sybase à une instruction SQL. Vous indiquez ce délai d'attente en secondes. Vous pouvez modifier les cinq définitions de serveur à la fois à l'aide de l'instruction ALTER SERVER suivante :

```
ALTER SERVER TYPE sybase  
OPTIONS (ADD FOLD_ID 'U', ADD TIMEOUT '600')
```

Utilisation d'options de serveur dans des définitions de serveur (Centre de contrôle DB2)

Il existe des options de serveur générales et des options de serveur réservées à certains types de sources de données. Vous pouvez modifier une définition de serveur à partir du Centre de contrôle ou de l'invite de ligne de commande DB2 pour ajouter ou modifier une option de serveur.

Avant de commencer

L'ID autorisation qui émet l'instruction ALTER SERVER doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Voir «Restrictions applicables à la modification de définitions de serveur», à la page 26.

A propos de cette tâche

Les options de serveur sont définies à des valeurs qui sont conservées au cours de connexions successives à la source de données. Ces valeurs sont stockées dans le catalogue du système fédéré.

Procédure

Pour effectuer cette tâche à partir du Centre de contrôle DB2 :

1. Développez le dossier Objets de base de données fédérée. Les objets de définition de serveur sont affichés dans le panneau du contenu de la fenêtre Centre de contrôle DB2.
2. Cliquez avec le bouton droit de la souris sur la définition de serveur que vous souhaitez modifier puis cliquez sur **Modification** dans la liste des actions. Le bloc-notes Modification d'une définition de serveur s'ouvre.
3. Dans la page Paramètres, sélectionnez l'option de serveur que vous souhaitez ajouter ou supprimer.
4. Indiquez la valeur de chaque option que vous ajoutez ou modifiez.
5. Cliquez sur **OK** pour modifier la définition de serveur et fermer le bloc-notes Modification d'une définition de serveur.

Certaines options de serveur sont obligatoires et ne peuvent pas être supprimées. D'autres ne peuvent pas être ajoutées si certaines options de serveur sont déjà définies.

Modification temporaire des options de serveur pour les sources de données relationnelles

L'instruction SET SERVER OPTION se substitue à la valeur de l'option de serveur dans la définition de serveur pour la durée d'une connexion unique à la base de données fédérée. La valeur de substitution n'est pas stockée dans le catalogue global.

A propos de cette tâche

Pour le SQL statique, l'utilisation de l'instruction SET SERVER OPTION affecte uniquement l'exécution de l'instruction SQL statique. L'utilisation de l'instruction SET SERVER OPTION n'a aucun effet sur les plans générés par l'optimisation.

Procédure

Pour définir temporairement une valeur d'option de serveur pour une source de données relationnelle, utilisez l'instruction SET SERVER OPTION.

Par exemple :

```
SET SERVER OPTION PLAN_HINTS TO Y' FOR SERVER SYB_SERVER
```

Hierarchie des paramètres des options de serveur

Lorsque vous avez la même option de serveur définie avec une valeur pour un type de source de données et avec une autre valeur sur un serveur de source de données particulier, une hiérarchie est établie entre les paramètres.

Par exemple, l'option de serveur PLAN_HINTS est définie sur 'Y' pour le type de source de données SYBASE. Cependant, elle est définie sur 'N' dans la définition de serveur d'un serveur de source de données Sybase particulier PURNELL. Le paramètre défini pour le serveur de source de données particulier prend le pas sur le paramètre défini pour le type de source de données. Cette configuration entraîne l'activation de PLAN_HINTS sur tous les serveurs de source de données Sybase, à l'exception de PURNELL.

Utilisation d'options de serveur dans des définitions de serveur (ligne de commande DB2)

Il existe des options de serveur générales et des options de serveur réservées à certains types de sources de données. Vous pouvez modifier une définition de serveur à partir du Centre de contrôle ou de l'invite de ligne de commande DB2 pour ajouter ou modifier une option de serveur.

Avant de commencer

L'ID autorisation qui émet l'instruction ALTER SERVER doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Voir «Restrictions applicables à la modification de définitions de serveur», à la page 26.

A propos de cette tâche

Les options de serveur sont définies à des valeurs qui sont conservées au cours de connexions successives à la source de données. Ces valeurs sont stockées dans le catalogue du système fédéré.

Procédure

Pour effectuer cette tâche à partir de l'invite de la ligne de commande, émettez l'instruction ALTER SERVER. Par exemple :

- Vous avez créé une définition de serveur pour un serveur Informix en utilisant le nom de serveur INFMX01. Vous souhaitez maintenant attribuer la valeur Y à l'option DB2_MAXIMAL_PUSHDOWN. L'instruction permettant de modifier la définition de serveur est la suivante :

```
ALTER SERVER INFMX01 OPTIONS (SET DB2_MAXIMAL_PUSHDOWN 'Y')
```

- Vous avez créé une définition de serveur pour un serveur Oracle en utilisant le nom de serveur ORCL99. Vous souhaitez maintenant ajouter les options FOLD_ID et FOLD_PW à cette définition. L'instruction permettant de modifier la définition de serveur est la suivante :

```
ALTER SERVER ORCL99 OPTIONS (ADD FOLD_ID 'U', FOLD_PW 'U')
```

- Vous souhaitez définir la valeur du délai d'attente au nombre de secondes pendant lesquelles l'encapsuleur CTLIB doit attendre une réponse du serveur Sybase. Vous utilisez l'option de serveur TIMEOUT pour définir cette valeur. L'instruction permettant de modifier la définition de serveur est la suivante :
`ALTER SERVER SYBSERVER OPTIONS (ADD TIMEOUT '60')`

Modification d'un mappage utilisateur (Centre de contrôle DB2)

Un mappage utilisateur est une association entre l'ID autorisation utilisé sur le serveur fédéré et l'ID autorisation employé sur la source de données. Les mappages utilisateur sont nécessaires pour que les requêtes réparties puissent être envoyées à la source de données.

Avant de commencer

Si l'ID autorisation qui émet l'instruction diffère de l'ID autorisation mappée sur la source de données, elle doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Le serveur fédéré ne peut pas traiter une instruction ALTER USER MAPPING dans une unité d'oeuvre donnée si cette dernière contient déjà l'une des instructions suivantes :

- une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue sur la source de données incluse dans le mappage ;
- un curseur ouvert sur un pseudonyme pour une table ou une vue sur la source de données incluse dans le mappage ;
- une insertion, suppression ou mise à jour émise pour un pseudonyme d'une table ou d'une vue sur la source de données incluse dans le mappage.

A propos de cette tâche

L'instruction ALTER USER MAPPING permet de remplacer l'ID autorisation ou le mot de passe utilisé sur la source de données par un ID autorisation de serveur fédéré particulier.

Vous pouvez modifier un mappage utilisateur à partir du Centre de contrôle DB2 ou de l'invite de ligne de commande.

Procédure

Pour modifier un mappage utilisateur à partir du Centre de contrôle DB2 :

1. Développez le dossier Objets de base de données fédérée. Les objets de mappage utilisateur sont affichés dans le panneau du contenu de la fenêtre Centre de contrôle DB2.
2. Cliquez avec le bouton droit de la souris sur le mappage utilisateur que vous souhaitez modifier puis cliquez sur **Modification** dans la liste des actions. La fenêtre Modification de mappage utilisateur s'ouvre.
3. Modifiez la valeur de l'option.
4. Cliquez sur **OK** pour modifier le mappage utilisateur et fermer le bloc-notes Modification de mappage utilisateur.

Modification d'un mappage utilisateur (ligne de commande DB2)

Un mappage utilisateur est une association entre l'ID autorisation utilisé sur le serveur fédéré et l'ID autorisation employé sur la source de données. Les mappages utilisateur sont nécessaires pour que les requêtes réparties puissent être envoyées à la source de données.

Avant de commencer

Si l'ID autorisation qui émet l'instruction diffère de l'ID autorisation mappée sur la source de données, elle doit inclure des droits d'accès SYSADM ou DBADM sur la base de données fédérée.

Restrictions

Le serveur fédéré ne peut pas traiter une instruction ALTER USER MAPPING dans une unité d'oeuvre donnée si cette dernière contient déjà l'une des instructions suivantes :

- une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue sur la source de données incluse dans le mappage ;
- un curseur ouvert sur un pseudonyme pour une table ou une vue sur la source de données incluse dans le mappage ;
- une insertion, suppression ou mise à jour émise pour un pseudonyme d'une table ou d'une vue sur la source de données incluse dans le mappage.

A propos de cette tâche

L'instruction ALTER USER MAPPING permet de remplacer l'ID autorisation ou le mot de passe utilisé sur la source de données par un ID autorisation de serveur fédéré particulier.

Vous pouvez modifier un mappage utilisateur à partir du Centre de contrôle DB2 ou de l'invite de ligne de commande.

Procédure

Pour modifier un mappage utilisateur à partir de la ligne de commande DB2, émettez l'instruction ALTER USER MAPPING :

Par exemple, Jenny utilise le serveur fédéré pour se connecter à un serveur Sybase appelé SYBSERVER. Elle accède au serveur fédéré avec l'ID autorisation *jennifer*. L'ID autorisation *jennifer* est mappé sur l'ID autorisation *jenn* sur le serveur Sybase. L'ID autorisation pour Jenny sur le serveur Sybase est remplacé par *jen123*. L'instruction ALTER USER MAPPING permettant de mapper *jennifer* sur *jen123* est :

```
ALTER USER MAPPING FOR jennifer SERVER SYBSERVER
  OPTIONS (SET REMOTE_AUTHID 'jen123')
```

Tomas utilise le serveur fédéré pour se connecter à un serveur Oracle appelé ORASERVER. Il accède au serveur fédéré avec l'ID autorisation *tomas*. L'ID autorisation *tomas* est mappé sur l'ID autorisation *tom* sur le serveur Oracle. Le mot de passe de Tomas sur le serveur Oracle est modifié. Son nouveau mot de passe est *day2night*. L'instruction ALTER USER MAPPING permettant de mapper *tomas* sur le nouveau mot de passe est :

```
ALTER USER MAPPING FOR tomas SERVER ORASERVER
OPTIONS (SET REMOTE_PASSWORD 'day2night')
```

Les options utilisateur REMOTE_AUTHID et REMOTE_PASSWORD respectent la distinction majuscules/minuscules sauf si vous définissez les options de serveur FOLD_ID et FOLD_PW à 'U' ou 'L' dans l'instruction CREATE SERVER.

Modification d'un pseudonyme (Centre de contrôle DB2)

Les pseudonymes sont des identificateurs utilisés pour référencer un objet auquel vous souhaitez accéder sur une source de données. Vous pouvez modifier les noms de colonne de la source de données qui sont stockés dans le catalogue global et définir des options de colonne en modifiant les pseudonymes. Vous pouvez modifier un pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Vous pouvez modifier un pseudonyme pour :

- modifier les noms locaux des colonnes de l'objet de source de données ;
- modifier les types de données locaux des colonnes de l'objet de source de données ;
- ajouter, définir ou supprimer des options de pseudonymes et de colonne
- ajouter ou supprimer une clé primaire ;
- ajouter ou supprimer une ou plusieurs contraintes uniques, référentielles ou de vérification ;
- modifier un ou plusieurs attributs de contrainte référentielle, de vérification ou de dépendance fonctionnelle ;
- empêcher la mise en cache des pseudonymes sur le serveur fédéré ;
- activer la mise en cache des pseudonymes sur le serveur fédéré. Si des tables de la mémoire cache ou des tables de requêtes matérialisées sont associées à un pseudonyme mis en cache, vous devez les supprimer pour pouvoir modifier l'option de mise en cache.

Procédure

Pour modifier un pseudonyme à partir du Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonyme**.

2. Cliquez avec le bouton droit de la souris sur le pseudonyme que vous souhaitez modifier puis cliquez sur **Modification**. Le bloc-notes Modification du pseudonyme s'ouvre.
3. Sur la page Pseudonyme, modifiez les options concernées.
4. Sur la page Clés, définissez les contraintes d'intégrité référentielle pour le pseudonyme. Vous pouvez définir une contrainte de clé primaire, de clé unique ou de clé externe.
5. Sur la page Contraintes de vérification, définissez les contraintes de vérification ou de dépendance fonctionnelle pour le pseudonyme.
6. Sur la page Paramètres, définissez les options du pseudonyme.
7. Cliquez sur **OK** pour modifier le pseudonyme et fermer le bloc-notes.
Certains options de pseudonymes sont obligatoires et ne peuvent pas être supprimées. D'autres ne peuvent pas être ajoutées si certaines options de pseudonymes sont déjà définies.

Lorsque la structure ou le contenu de l'objet de source de données change de façon significative, vous devez mettre à jour les statistiques de pseudonymes. Les modifications significatives comprennent l'ajout ou la suppression de plusieurs lignes.

Restrictions pour les modifications de pseudonymes

Lorsque vous modifiez un pseudonyme, certaines restrictions s'appliquent.

Options de colonne

Si l'une des options suivantes est définie sur une colonne, vous ne pouvez ajouter aucune autre option à cette colonne :

- SOAPACTIONCOLUMN
- URLCOLUMN
- PRIMARY_KEY
- FOREIGN_KEY

Pour BioRS

- Si vous modifiez le nom d'élément d'une colonne en utilisant l'option `ELEMENT_NAME`, aucune vérification n'est effectuée sur le nouveau nom pour contrôler sa validité. Une option incorrecte peut entraîner des erreurs lorsque la colonne est référencée dans une requête.
- Si vous apportez des modifications à l'option de colonne `IS_INDEXED`, ces modifications ne sont pas vérifiées auprès du serveur BioRS. Une option incorrecte peut entraîner des erreurs lorsque la colonne est référencée dans une requête.

Types de données

- Si vous modifiez le type de données d'une colonne, le nouveau type doit être compatible avec le type de la colonne ou de l'élément de source de données correspondant. Le remplacement du type de données local par un type incompatible avec le type de données distant peut provoquer des erreurs imprévisibles.
- La valeur `type_données_local` ne peut pas être `LONG VARCHAR`, `LONG VARGRAPHIC`, `XML`, ni un type de données défini par l'utilisateur.
- La valeur `type_données_source_données` ne peut pas être un type défini par l'utilisateur.
- Vous ne pouvez pas remplacer les types locaux existants ni en créer de nouveaux pour certaines des sources de données non relationnelles.

Consultez la documentation de l'encapsuleur de source de données particulier pour plus d'informations sur cette restriction.

- Lorsque la spécification locale du type de données d'une colonne est modifiée, le gestionnaire de la base de données fédérée invalide les statistiques (par exemple, HIGH2KEY et LOW2KEY) qui sont recueillies pour cette colonne.
- Le type local est défini pour l'objet de source de données particulier lorsque des utilisateurs y accèdent avec ce pseudonyme. Le même objet de source de données peut avoir un autre pseudonyme qui utilise le mappage de type de données par défaut.

Index Il est impossible d'utiliser l'instruction ALTER NICKNAME pour enregistrer un nouvel index de source de données dans la base de données fédérée. Utilisez l'instruction CREATE INDEX avec la clause SPECIFICATION ONLY pour créer une spécification d'index.

Paramètres LOCAL NAME et LOCAL TYPE

- L'instruction ALTER NICKNAME ne peut pas être utilisée pour modifier les noms locaux ni les types de données pour les colonnes du pseudonyme si :
 - Le pseudonyme est utilisé dans une vue, une méthode SQL ou une fonction SQL ;
 - vous définissez une contrainte informationnelle sur le pseudonyme ;
- la clause federated_column_options doit être spécifiée en dernier si vous devez également spécifier le paramètre LOCAL NAME, le paramètre LOCAL TYPE ou les deux dans l'instruction ALTER NICKNAME.

Pseudonyme

Il est impossible d'utiliser l'instruction ALTER NICKNAME pour modifier le nom de la banque de données BioRS référencée par ou utilisée dans un pseudonyme BioRS. Si le nom d'une banque de données BioRS change, vous devez supprimer le pseudonyme et le recréer.

Vous ne pouvez pas utiliser l'instruction ALTER NICKNAME pour désactiver la mise en cache sur un pseudonyme avec des tables de mémoire cache ou des tables de requêtes matérialisées. Vous devez supprimer les tables de mémoire cache et les tables de requêtes matérialisées avant de désactiver la mise en cache sur le pseudonyme.

Unités d'oeuvre

Le serveur fédéré ne peut pas traiter une instruction ALTER NICKNAME dans une unité d'oeuvre donnée lorsque l'une des conditions suivantes se produit :

- si un curseur est ouvert dans la même unité d'oeuvre sur le pseudonyme référencé dans l'instruction ALTER NICKNAME ;
- si une insertion, une suppression ou une mise à jour est effectuée dans la même unité d'oeuvre pour le pseudonyme référencé dans l'instruction ALTER NICKNAME.

Modification des noms de colonne de pseudonymes (Centre de contrôle DB2)

Vous pouvez modifier un pseudonyme pour changer les noms des colonnes. Vous pouvez modifier des noms de colonne à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Lorsque vous créez un pseudonyme, les noms de colonne qui sont associés à l'objet source de données sont stockés dans la base de données fédérée. Pour certaines sources, l'encapsuleur indique les noms de colonne. Pour d'autres sources, vous devez indiquer ces noms en créant le pseudonyme.

Procédure

Pour modifier des noms de colonne de pseudonymes à partir du Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonyme**.
2. Cliquez avec le bouton droit de la souris sur le pseudonyme que vous souhaitez modifier puis cliquez sur **Modification**. Le bloc-notes Modification du pseudonyme s'ouvre.
3. Dans la page Pseudonyme, sélectionnez la colonne que vous souhaitez modifier puis cliquez sur **Modification**. La fenêtre Modification de la colonne s'ouvre.
4. Entrez le nom de la colonne.
5. Cliquez sur **OK** pour changer le nom de la colonne et fermer la fenêtre.
6. Cliquez sur **OK** pour modifier le pseudonyme et fermer le bloc-notes.

Modification des noms de colonne de pseudonymes (ligne de commande DB2)

Vous pouvez modifier un pseudonyme pour changer les noms des colonnes. Vous pouvez modifier des noms de colonne à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe

- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Lorsque vous créez un pseudonyme, les noms de colonne qui sont associés à l'objet source de données sont stockés dans la base de données fédérée. Pour certaines sources, l'encapsuleur indique les noms de colonne. Pour d'autres sources, vous devez indiquer ces noms en créant le pseudonyme.

Procédure

Pour modifier des noms de colonne de pseudonymes à partir de la ligne de commande DB2, émettez l'instruction ALTER NICKNAME.

```
ALTER NICKNAME pseudonyme
ALTER COLUMN nom_actuel
LOCAL NAME nouveau_nom
```

Modification des options de pseudonymes (Centre de contrôle DB2)

Les options de pseudonymes sont des paramètres que vous indiquez sur le pseudonyme lorsque vous émettez les instructions CREATE NICKNAME et ALTER NICKNAME. Vous pouvez ajouter, définir ou supprimer des options de pseudonymes à l'aide de l'instruction ALTER NICKNAME. Vous pouvez modifier des noms de colonne à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Par exemple, le pseudonyme DRUGDATA1 est créé pour le fichier structuré par tableaux drugdata1.txt. Le chemin qualifié complet défini à l'origine dans l'instruction CREATE NICKNAME était /user/pat/drugdata1.txt.

Pour modifier l'option de pseudonymes FILE_PATH, émettez l'instruction suivante :

Procédure

Pour modifier des noms de colonne à partir du Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonyme**.
2. Cliquez avec le bouton droit de la souris sur le pseudonyme que vous souhaitez modifier puis cliquez sur **Modification**. Le bloc-notes Modification du pseudonyme s'ouvre.
3. Dans la page Paramètres, cochez la case en regard des options que vous souhaitez ajouter ou supprimer. Vous ne pouvez pas supprimer une option obligatoire.
4. Pour définir ou modifier la valeur d'une option, cliquez sur la zone **Valeur** de cette option. Selon l'option, vous avez le choix entre sélectionner une valeur dans la liste, cliquer pour sélectionner plusieurs valeurs ou entrer une nouvelle valeur.
5. Cliquez sur **OK** pour modifier le pseudonyme et fermer le bloc-notes.

Modification des options de pseudonymes (ligne de commande DB2)

Les options de pseudonymes sont des paramètres que vous indiquez sur le pseudonyme lorsque vous émettez les instructions CREATE NICKNAME et ALTER NICKNAME. Vous pouvez ajouter, définir ou supprimer des options de pseudonymes à l'aide de l'instruction ALTER NICKNAME. Vous pouvez modifier les options de pseudonymes à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

Procédure

Pour modifier une option de pseudonymes à partir de l'invite de la ligne de commande, émettez l'instruction ALTER NICKNAME.

```
ALTER NICKNAME pseudonyme  
  OPTIONS (SET nom_option 'valeur_chaine_option')
```

Exemple : Le pseudonyme DRUGDATA1 est créé pour le fichier structuré par tableaux drugdata1.txt. Le chemin qualifié complet défini à l'origine dans l'instruction CREATE NICKNAME était /user/pat/drugdata1.txt. Pour modifier l'option de pseudonymes FILE_PATH, émettez l'instruction suivante :

```
ALTER NICKNAME DRUGDATA1 OPTIONS (SET FILE_PATH '/usr/kelly/data/drugdata1.txt')
```

Modification des options de colonne de pseudonymes (Centre de contrôle DB2)

Vous pouvez ajouter, définir ou supprimer des options de colonne de pseudonymes à l'aide de l'instruction ALTER NICKNAME. Vous pouvez modifier des noms de colonne à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Vous indiquez des informations relatives aux colonnes dans les instructions CREATE NICKNAME et ALTER NICKNAME à l'aide de paramètres appelés options de colonne de pseudonymes.

Procédure

Pour modifier des options de colonne de pseudonymes à partir du Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonyme**.
2. Cliquez avec le bouton droit de la souris sur le pseudonyme que vous souhaitez modifier puis cliquez sur **Modification**. Le bloc-notes Modification du pseudonyme s'ouvre.
3. Dans la page Pseudonyme, sélectionnez la colonne que vous souhaitez modifier puis cliquez sur **Modification**. La fenêtre Modification de la colonne s'ouvre.
4. Sélectionnez l'option de colonne que vous souhaitez ajouter ou supprimer.
5. Indiquez la valeur de chaque option que vous ajoutez ou modifiez.
6. Cliquez sur **OK** pour changer l'option de la colonne et fermer la fenêtre.
7. Cliquez sur **OK** pour modifier le pseudonyme et fermer le bloc-notes.

Modification des options de colonne de pseudonymes (ligne de commande DB2)

Vous pouvez ajouter, définir ou supprimer des options de colonne de pseudonymes à l'aide de l'instruction ALTER NICKNAME. Vous pouvez modifier des noms de colonne à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Vous indiquez des informations relatives aux colonnes dans les instructions CREATE NICKNAME et ALTER NICKNAME à l'aide de paramètres appelés options de colonne de pseudonymes. Vous pouvez indiquer l'une de ces valeurs en majuscules ou en minuscules.

Procédure

Pour modifier des noms de colonne de pseudonymes à partir de l'invite de ligne de commande, utilisez l'instruction ALTER NICKNAME.

Exemple 1 : Définition de l'option de colonne NUMERIC_STRING avec des sources de données relationnelles

L'option de colonne NUMERIC_STRING s'applique à des colonnes de type caractères (CHAR and VARCHAR). Supposez qu'une source de données possède une séquence de classement différente de celle de la base de données fédérée. Le serveur fédéré ne trie généralement pas les colonnes contenant des données de type caractères sur la source de données. Il renvoie plutôt ces données à la base de données fédérée pour effectuer le tri localement. Cependant, supposez que la colonne est un type de données caractères et contient uniquement des caractères numériques ('0','1',..., '9'). Vous pouvez l'indiquer en affectant la valeur 'Y' à l'option de colonne NUMERIC_STRING. Cela permet à l'optimiseur de requêtes DB2 UDB d'effectuer le tri sur la source de données. Si le tri est effectué à distance, vous pouvez éviter le temps de traitement nécessaire pour le tri des données sur le serveur fédéré.

Le pseudonyme ORA_INDSALES désigne une table Oracle appelée INDONESIA_SALES. La table contient la colonne POSTAL_CODE avec le type de données VARCHAR. A l'origine, la colonne ne contenait que des caractères numériques, et la valeur 'Y' était affectée à l'option de colonne

NUMERIC_STRING. Cependant, à présent la colonne mêle caractères numériques et non numériques. Pour remplacer la valeur de l'option de colonne NUMERIC_STRING par 'N', utilisez l'instruction suivante :

```
ALTER NICKNAME ORA_INDSALES ALTER COLUMN POSTAL_CODE
  OPTIONS (SET NUMERIC_STRING 'N')
```

Exemple 2 : Définition de l'option de colonne VARCHAR_NO_TRAILING_BLANKS avec des sources de données relationnelles

L'option de colonne VARCHAR_NO_TRAILING_BLANKS peut être utilisée pour identifier des colonnes particulières ne contenant aucun blanc de fin. Le compilateur SQL Compiler prend en compte ce paramètre pour vérifier toutes les opérations (comme les opérations de comparaison) effectuées sur les colonnes.

Le pseudonyme ORA_INDSALES désigne une table Oracle appelée INDONESIA_SALES. La table contient la colonne NAME avec le type de données VARCHAR. La colonne NAME ne contient pas de blancs de fin. Pour ajouter l'option VARCHAR_NO_TRAILING_BLANKS au pseudonyme, utilisez l'instruction suivante :

```
ALTER NICKNAME ORA_INDSALES ALTER COLUMN NAME
  OPTIONS (ADD VARCHAR_NO_TRAILING_BLANKS 'Y')
```

Exemple 3 : Définition de l'option de colonne XPATH avec des sources de données non relationnelles

Le pseudonyme EMPLOYEE désigne une source de données XML. Un chemin XPATH a été défini pour la colonne *nomf*. Pour définir un autre chemin pour l'option de colonne XPATH, utilisez l'instruction suivante :

```
ALTER NICKNAME EMPLOYEE ALTER COLUMN nomf
  OPTIONS (SET XPATH './@first')
```

Modification d'un pseudonyme (ligne de commande DB2)

Les pseudonymes sont des identificateurs utilisés pour référencer un objet auquel vous souhaitez accéder sur une source de données. Vous pouvez modifier les noms de colonne de la source de données qui sont stockés dans le catalogue global et définir des options de colonne en modifiant les pseudonymes. Vous pouvez modifier un pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Vous pouvez modifier un pseudonyme pour :

- modifier les noms locaux des colonnes de l'objet de source de données ;
- modifier les types de données locaux des colonnes de l'objet de source de données ;
- ajouter, définir ou supprimer des options de pseudonymes et de colonne
- ajouter ou supprimer une clé primaire ;
- ajouter ou supprimer une ou plusieurs contraintes uniques, référentielles ou de vérification ;
- modifier un ou plusieurs attributs de contrainte référentielle, de vérification ou de dépendance fonctionnelle ;

Procédure

Pour modifier un pseudonyme à partir de la ligne de commande DB2, émettez l'instruction ALTER NICKNAME avec les paramètres appropriés définis.

Lorsque la structure ou le contenu de l'objet de source de données change de façon significative, vous devez mettre à jour les statistiques de pseudonymes. Les modifications significatives comprennent l'ajout ou la suppression de plusieurs lignes.

Suppression d'un encapsuleur

La suppression d'un encapsuleur peut s'avérer nécessaire pour plusieurs raisons.

Avant de commencer

Pour émettre l'instruction DROP WRAPPER, vous devez disposer de droits d'accès SYSADM ou DBADM.

A propos de cette tâche

Parfois, vous pouvez utiliser plusieurs encapsuleurs pour accéder à une source de données. Celui que vous choisissez dépend de la version du logiciel client de source de données que vous utilisez. Ou bien, il peut dépendre du système d'exploitation que vous utilisez sur le serveur fédéré. Supposons que vous souhaitiez accéder à deux tables et une vue Oracle. Vous utilisez Oracle version 10 et le système d'exploitation du serveur fédéré est Windows. Au départ, vous avez créé l'encapsuleur SQLNET. Comme IBM InfoSphere Federation Server ne prend pas en charge l'encapsuleur SQLNET, vous pouvez supprimer ce dernier et créer l'encapsuleur NET8.

Une autre raison de supprimer un encapsuleur est que vous n'avez plus besoin d'accéder à la source de données à laquelle l'encapsuleur est associé. Par exemple, supposez que votre entreprise doive accéder à des informations client dans des bases de données SQL Informix et Microsoft SQL Server. Vous créez un encapsuleur pour la source de données Informix et un pour la source de données Microsoft SQL Server. Par la suite, votre entreprise décide de faire migrer toutes les informations depuis Microsoft SQL Server vers Informix. Vous n'avez alors plus besoin de l'encapsuleur Microsoft SQL Server et vous pouvez le supprimer.

Important : La suppression d'un encapsuleur a d'importantes répercussions. En effet, d'autres objets enregistrés auprès du serveur fédéré subissent des conséquences :

- toutes les définitions de serveur qui dépendent de l'encapsuleur supprimé sont également supprimés ;
- tous les objets qui dépendent des définitions de serveur supprimées sont également supprimés ;
- tous les pseudonymes qui dépendent des définitions de serveur supprimées sont également supprimés ; la suppression des pseudonymes qui dépendent de la définition de serveur influe sur les objets qui en dépendent :
 - les spécifications d'index qui dépendent des pseudonymes supprimés sont également supprimées ;
 - les vues qui dépendent des pseudonymes supprimés sont marquées comme inopérantes ;
 - les tables de requêtes matérialisées qui dépendent des pseudonymes supprimés sont également supprimées ;
- tous les modules et les instructions SQL dynamiques mises en cache qui dépendent des pseudonymes supprimés sont marqués comme non valides et le restent jusqu'à ce que les objets dépendants soient recréés.

Procédure

Pour supprimer un encapsuleur, utilisez l'instruction DROP.

Exemple : Supprimez l'encapsuleur Microsoft SQL Server *MSSQLODBC3* :

```
DROP WRAPPER MSSQLODBC3
```

Suppression d'une définition de serveur

La suppression d'une définition entraîne l'élimination de cette définition du catalogue global. L'objet de source de données auquel la définition de serveur fait référence ne subit aucune modification. Vous pouvez supprimer une définition de serveur à partir du Centre de contrôle DB2 ou de l'interpréteur de commande DB2.

Avant de commencer

Vous devez disposer de droits d'accès SYSADM ou DBADM pour supprimer une définition de serveur.

Restrictions

Le serveur fédéré ne peut pas traiter une instruction DROP SERVER dans une unité d'oeuvre donnée lorsque l'une des conditions suivantes se produit :

- l'instruction fait référence à une source de données unique et l'unité d'oeuvre contient déjà l'une des instructions suivantes :
 - une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue dans la source de données ;
 - un curseur ouvert sur un pseudonyme pour une table ou une vue dans la source de données ;
 - une insertion, suppression ou mise à jour émise sur un pseudonyme d'une table ou d'une vue dans la source de données ;

- L'instruction fait référence à une catégorie de sources de données (par exemple, toutes les sources de données d'un type et d'une version particulières) et l'unité d'oeuvre contient l'une des instructions suivantes :
 - une instruction SELECT faisant référence à un pseudonyme pour une table ou une vue dans l'une des sources de données ;
 - un curseur ouvert sur un pseudonyme pour une table ou une vue dans l'un des sources de données ;
 - une insertion, suppression ou mise à jour émise sur un pseudonyme d'une table ou d'une vue dans l'une des sources de données.

A propos de cette tâche

Lorsque vous n'avez plus besoin d'accéder à un serveur de source de données, supprimez la définition de serveur de la base de données fédérée. La suppression d'une définition de serveur a des conséquences sur d'autres objets enregistrés auprès du serveur fédéré :

- tous les mappages de fonctions définies par l'utilisateur, les mappages de types de données définis par l'utilisateur et les mappages utilisateur qui dépendent de la définition de serveur supprimée sont également supprimés ;
- tous les pseudonymes qui dépendent de la définition de serveur supprimée sont également supprimés ; la suppression des pseudonymes qui dépendent de la définition de serveur influe sur les objets qui en dépendent :
 - les spécifications d'index qui dépendent des pseudonymes supprimés sont également supprimées ;
 - les vues qui dépendent des pseudonymes supprimés sont marquées comme inopérantes ;
 - les tables de requêtes matérialisées qui dépendent des pseudonymes supprimés sont également supprimées ;
- tous les modules et les instructions SQL dynamiques mises en cache qui dépendent des pseudonymes supprimés sont marqués comme non valides et le restent jusqu'à ce que les objets dépendants soient recréés.

Procédure

Pour supprimer une définition de serveur, émettez l'instruction DROP :

```
DROP SERVER nom_serveur
```

où *nom_serveur* désigne la définition de serveur à supprimer.

Exemple : Un serveur Informix utilise le nom de serveur INFMX01. L'instruction DROP suivante supprime la définition de serveur :

```
DROP SERVER INFMX01
```

Suppression d'un mappage utilisateur

Lorsqu'un utilisateur n'a plus besoin d'accéder à une source de données distante, supprimez le mappage utilisateur entre le serveur fédéré et le serveur de source de données distant. Si l'utilisateur est mappé avec plusieurs serveurs de source de données, vous devez supprimer chaque mappage séparément.

Avant de commencer

Pour émettre l'instruction DROP USER MAPPING, l'ID autorisation de l'instruction DROP doit disposer du droit d'accès SYSADM ou DBADM, si cet ID autorisation est différent de l'ID utilisateur de la base de données fédérée indiqué dans le mappage utilisateur. Sinon, si l'ID autorisation et l'ID utilisateur du mappage utilisateur concordent, aucun droit d'accès ni privilège n'est requis.

Procédure

Pour supprimer un mappage utilisateur, émettez l'instruction DROP :

```
DROP USER MAPPING FOR ID_utilisateur SERVER nom_serveur_local
```

où :

- *ID_utilisateur* est l'ID autorisation de l'utilisateur sur le serveur fédéré ;
- *nom_serveur_local* est le nom local servant à identifier le serveur source de données distant dans la définition de serveur.

Suppression d'un pseudonyme

La suppression d'un pseudonyme entraîne son élimination du catalogue global sur le serveur fédéré. L'objet de source de données auquel le pseudonyme fait référence ne subit aucune modification.

Avant de commencer

Le pseudonyme doit être répertorié dans le catalogue.

L'ID autorisation de l'instruction DROP doit détenir l'un des privilèges suivants lors de la suppression de pseudonymes :

- droit SYSADM ou DBADM
- privilège DROPIN sur le schéma pour le pseudonyme
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme
- privilège CONTROL sur le pseudonyme

Restrictions

Pour les pseudonymes renvoyant à des sources de données relationnelles, le serveur fédéré ne peut pas traiter l'instruction DROP NICKNAME dans une unité d'oeuvre donnée lorsque l'une des conditions suivantes se produit :

- un curseur est ouvert dans la même unité d'oeuvre sur le pseudonyme référencé dans l'instruction ;
- un pseudonyme référencé dans cette instruction est déjà référencé par une instruction SELECT dans la même unité de travail ;
- une insertion, une suppression ou une mise à jour est effectuée dans la même unité d'oeuvre pour le pseudonyme référencé dans l'instruction.

Pour les pseudonymes renvoyant à des sources de données non relationnelles, le serveur fédéré ne peut pas traiter l'instruction DROP NICKNAME dans une unité d'oeuvre donnée lorsque l'une des conditions suivantes se produit :

- un curseur est ouvert dans la même unité d'oeuvre sur un pseudonyme référencé dans cette instruction ;
- un pseudonyme référencé dans cette instruction est déjà référencé par une instruction SELECT dans la même unité de travail ;

A propos de cette tâche

La suppression d'un pseudonyme a des conséquences sur d'autres objets enregistrés auprès du serveur fédéré :

- la suppression d'un pseudonyme influe sur les objets qui en dépendent :
 - les spécifications d'index qui dépendent des pseudonymes supprimés sont également supprimés ;
 - les vues qui dépendent des pseudonymes supprimés sont marquées comme inopérantes ;
 - les tables de requêtes matérialisées qui dépendent des pseudonymes supprimés sont également supprimés ;
- tous les modules et les instructions SQL dynamiques mises en cache qui dépendent des pseudonymes supprimés sont marqués comme non valides et le restent jusqu'à ce que les objets dépendants soient recréés.

Procédure

Pour supprimer un pseudonyme, émettez l'instruction DROP :

```
DROP NICKNAME pseudonyme
```

où *pseudonyme* désigne le pseudonyme à supprimer.

Chapitre 3. Mappages de types de données dans un système fédéré

Les types de données sur une source de données doivent être mappés aux types de données DB2 correspondants. Ce mappage permet au serveur fédéré d'extraire des données de la source de données.

La base de données fédérée fournit un ensemble de mappages de types de données par défaut pour certaines sources de données. Pour d'autres, vous devez fournir les mappages de types de données que vous souhaitez utiliser. Pour les sources de données non relationnelles, vous ne pouvez pas remplacer les mappages de types de données existants ni en créer de nouveaux.

Voici des exemples de mappages de types de données par défaut :

- Le type FLOAT Oracle est mappé par défaut au type DOUBLE DB2.
- Le type DATE Oracle est mappé par défaut au type TIMESTAMP DB2.
- Le type DATE DB2 for z/OS mappe par défaut vers la DATE de type DB2.

Les pseudonymes créés après la modification d'un mappage utilisent le nouveau mappage de type. Les pseudonymes créés avant la modification du mappage utilisent le nouveau mappage de type de données par défaut.

Si vous avez déjà créé les pseudonymes, vous pouvez mettre à jour les pseudonymes existants de l'une des façons suivantes :

- en modifiant chaque pseudonyme ;
- en supprimant puis recréant chaque pseudonyme.

Les serveurs fédérés DB2 ne prennent pas en charge les mappages pour les types de données suivants :

- le type de données local ne peut pas être LONG VARCHAR, LONG VARGRAPHIC, ni un type de données défini par l'utilisateur ;
- le type de données distant ne peut pas être un type défini par l'utilisateur.

Cependant, vous pouvez utiliser une fonction de transtypage pour convertir le type de données défini par l'utilisateur dans une vue sur la source de données distante en type de données intégré au système. Vous pouvez alors créer un pseudonyme pour la vue. Pour la plupart des sources de données, si vous créez des vues de ce type, elles ne comportent ni statistiques ni index et ne peuvent pas être mises à jour.

Mappages de types de données et catalogue global de bases de données fédérées

Les définitions de types de données locales sont stockées dans la vue catalogue SYSCAT.COLUMNS du catalogue global de la base de données fédérée.

Lorsque vous écrivez une instruction CREATE NICKNAME, vous désignez un objet de source de données que le pseudonyme représente. Dans la plupart des cas, le serveur fédéré définit un type de données pris en charge par DB2 pour chaque colonne ou zone dans cet objet de source de données. Pour certaines sources de données non relationnelles, vous devez fournir le type de données DB2.

Pour les sources de données relationnelles, afin de déterminer le type de données locales à stocker dans la vue catalogue SYSCAT.COLUMNS, le serveur fédéré recherche des informations de mappage de type de données aval dans les encapsuleurs et dans la vue catalogue SYSCAT.TYPEMAPPINGS. Les mappages de cette vue prennent le pas sur les mappages par défaut des encapsuleurs. Si vous créez d'autres mappages pour remplacer les mappages de types de données par défaut, le serveur fédéré utilise ces autres mappages. Si plusieurs mappages s'appliquent à une colonne, le serveur fédéré utilise les mappages créés le plus récemment.

Pour les sources de données non relationnelles, afin de déterminer le type de données locales à stocker dans la vue catalogue SYSCAT.COLUMNS, le serveur fédéré recherche des informations de mappage de type de données aval dans les encapsuleurs. Selon la source de données non relationnelles, le degré de modification qu'il est possible d'apporter aux types de données définis par l'encapsuleur varie. Pour certaines sources de données non relationnelles, vous n'indiquez aucune colonne. L'encapsuleur définit les types de données. Pour d'autres sources, vous pouvez remplacer les types de données. Et pour d'autres sources encore, vous devez indiquer les types de données de colonne dans l'instruction CREATE NICKNAME.

Lorsque vous écrivez l'instruction DDL transparent CREATE TABLE pour des sources de données relationnelles, indiquez des types de données DB2 dans l'instruction. Le serveur fédéré recherche des informations sur les mappages de types de données inversés entre la base de données fédérée et la source de données. Le serveur fédéré recherche ces informations dans l'encapsuleur et dans la vue catalogue SYSCAT.TYPEMAPPINGS.

Lorsque des valeurs provenant d'une colonne de source de données sont renvoyées à la base de données fédérée, elles se conforment entièrement au type de données DB2 auquel la colonne de source de données est mappée. Si ce mappage est un mappage par défaut, les valeurs sont également entièrement conformes au type de source de données du mappage. Par exemple, si une table Oracle avec une colonne FLOAT est définie auprès de la base de données fédérée, le mappage par défaut du type Oracle FLOAT au type DB2 DOUBLE s'applique automatiquement à cette colonne. Les valeurs renvoyées par la colonne sont entièrement conformes aux types de données FLOAT et DOUBLE.

Quand créer des mappages de types de données alternatifs

Vous pouvez créer des mappages de types de données alternatives pour des sources de données relationnelles.

Il peut s'avérer nécessaire de créer des mappages de types de données alternatifs dans les cas suivants :

- Pour remplacer un mappage de types de données par défaut

Pour certains encapsuleurs, vous pouvez modifier le format ou la longueur des valeurs renvoyées. Pour ce faire, vous devez modifier le type de données DB2 auquel les valeurs doivent se conformer. Ainsi, le type de données Oracle DATE est utilisé comme horodatage et contient le siècle, l'année, le mois, le jour, l'heure, la minute et la seconde. Par défaut, ce type de données est mappé sur le type de données DB2 TIMESTAMP. Pour ne retourner que l'heure, la minute et la seconde, vous pouvez remplacer le mappage de types de données par défaut, afin que le type de données Oracle DATE soit mappé sur le type de données DB2 TIME. Lorsque les colonnes Oracle DATE font l'objet d'une requête, seule la partie horaire des valeurs d'horodatage Oracle est renvoyée au serveur fédéré.

- Lorsqu'il n'existe pas de mappage par défaut

Si un mappage de types de données par défaut n'est pas disponible pour un type de données de sources de données, vous devez en créer un pour le nouveau type de données.

Vous utilisez pour cela l'instruction CREATE TYPE MAPPING. Les mappages que vous créez sont stockés dans la vue catalogue SYSCAT.TYPEMAPPINGS de la base de données fédérée.

Data type mappings for nonrelational data sources

Pour certaines sources de données non relationnelles, les mappages de types de données ne figurent pas dans les encapsuleurs. Dans certains cas, vous devez indiquer les informations relatives au type local dans l'instruction CREATE NICKNAME.

L'exemple suivant indique comment les types de données de colonne sont spécifiés dans l'instruction CREATE NICKNAME pour certaines des sources de données non relationnelles :

```
CREATE NICKNAME DRUGDATA1
(Dcode Integer NOT NULL, Drug CHAR(20), Manufacturer CHAR(20))
FOR SERVER biochem_lab
OPTIONS (FILE_PATH '/usr/pat/DRUGDATA1.TXT', COLUMN_DELIMITER ',',
SORTED 'Y', KEY_COLUMN 'DCODE', VALIDATE_DATA_FILE 'Y')
```

Mappages de types de données avals et inversés

Les mappages de type aval et inversé sont les deux types de mappage entre types de données de sources de données et types de données de bases de données fédérées.

Un *mappage de type aval* est un mappage depuis un type de données distant vers un type de données local équivalent. Les mappages de type aval sont utilisés lorsque vous créez un pseudonyme pour un objet de source de données. Le type de données local équivalent pour chaque colonne de l'objet de source de données est stocké dans le catalogue global.

Un *mappage de type inversé* est un mappage depuis un type de données local vers un type de données distant équivalent. Les mappages de type inversé sont utilisés avec le langage DDL transparent.

La figure 2 présente des mappages de types de données aval et inversé.

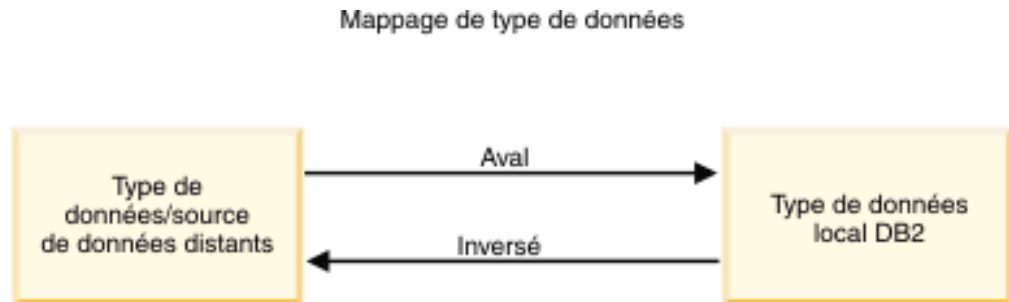


Figure 2. Mappages de types de données avals et inversés

Création de mappages de types de données

Utilisez l'instruction `CREATE TYPE MAPPING` pour créer un mappage de type de données. Vous pouvez exécuter cette instruction à partir du Centre de contrôle DB2 ou de l'interpréteur de commande, ou bien l'inclure dans un programme d'application. Vous ne pouvez pas utiliser le Centre de contrôle DB2 pour créer ou modifier des mappages de types de données.

Avant de commencer

L'ID autorisation associé à l'instruction doit détenir le droit d'accès `SYSADM` ou `DBADM`.

A propos de cette tâche

Voir `CREATE TYPE MAPPING (instruction)` pour davantage d'informations sur son utilisation.

Restrictions

- La valeur `type_données_local` ne peut pas être `LONG VARCHAR`, `LONG VARGRAPHIC`, ni un type de données défini par l'utilisateur.
- La valeur `type_données_source_données` ne peut pas être un type défini par l'utilisateur.
- Pour les sources de données non relationnelles, les capacités de substitution de mappages de types de données existants ou de création de mappages sont limitées.

Procédure

Exécutez l'instruction `CREATE TYPE MAPPING` pour créer un mappage de type de données.

Pour définir un type de serveur dans l'instruction CREATE TYPE MAPPING, vous devez utiliser l'une des valeurs suivantes pour *type-serveur* :

Tableau 4. Types de serveur valides

Source de données	Type de serveur
DB2 Database pour Linux, UNIX et Windows	DB2/CS DB2/UDB DB2/NT DB2/SUN DB2/HP DB2/HPUX DB2/AIX DB2/6000 DB2/PE DB2/PTX DB2/SCO DB2/LINUX DB2/EEE DB2/2
DB2 for z/OS	DB2/MVS DB2/ZOS DB2/390
DB2 Server for VSE et VM	DB2/VM DB2/VSE SQL/DS
DB2 for System i	DB2/400 DB2/ISERIES
Oracle	ORACLE
Informix	INFORMIX
ODBC	ODBC
Microsoft SQL Server	MSSQLSERVER
JDBC	JDBC
Teradata	TERADATA
Sybase	SYBASE

Création d'un mappage de type pour un type de données de source de données - exemple

Dans cet exemple, toutes les tables et vues Oracle qui utilisent le type de données Oracle NUMBER doivent être mappées sur le type de données DB2 DECIMAL(8,2). Le type de données Oracle NUMBER est mappé par défaut sur le type de données DB2, un type de données décimal à virgule flottante.

Utilisez l'instruction ALTER NICKNAME pour modifier les types locaux de pseudonymes existants. Vous devez modifier chaque pseudonyme séparément pour modifier le type de données local en DECIMAL(8,2). Si les pseudonymes n'existent pas, créez un mappage de type de données qui spécifie le type de source de données. Assurez-vous que l'encapsuleur de source de données a été créé avant d'exécuter l'instruction CREATE TYPE MAPPING. Pour créer le mappage du type de données Oracle NUMBER sur le type de données DB2 DECIMAL(8,2), exécutez l'instruction CREATE TYPE MAPPING, par exemple :

```
CREATE TYPE MAPPING MY_ORACLE_DEC FROM SYSIBM.DECIMAL(8,2)
TO SERVER TYPE ORACLE TYPE NUMBER
```

MY_ORACLE_DEC

Nom que vous attribuez au mappage de type. Ce nom ne peut pas dupliquer un nom de mappage de type de données qui existe déjà dans le catalogue.

FROM SYSIBM.DECIMAL(8,2)

Schéma DB2 local et type de données local. Si la longueur ou la précision et l'échelle ne sont pas spécifiés, ces valeurs sont déterminées à partir du type de données source.

TO SERVER TYPE ORACLE

Type de la source de données.

TYPE NUMBER

Type de données d'une source de données que vous mappez sur le type de données local. Les types de données définis par l'utilisateur ne sont pas autorisés.

Le type de données DB2 DECIMAL(8,2) est défini localement pour les colonnes Oracle. Lorsque vous créez des pseudonymes sur des tables et vues Oracle qui contiennent des colonnes NUMBER, le type de données Oracle NUMBER est mappé sur le type de données DB2 DECIMAL(8,2).

Création d'un mappage de type pour un type de données et une version de source de données - exemple

Dans cet exemple, il existe des tables et vues Oracle sur des versions différentes du serveur Oracle. Pour toutes les tables et vues sur des serveurs Oracle version 8.0.3, les colonnes qui utilisent le type de données Oracle NUMBER(23,3) doivent être mappées sur le type de données DB2 DECIMAL(8,2).

Le type de données Oracle NUMBER(23,3) est mappé par défaut sur le type de données DB2 DECIMAL(23,3). Utilisez l'instruction ALTER NICKNAME pour modifier les types locaux de pseudonymes existants. Vous devez modifier chaque pseudonyme séparément pour modifier le type de données local en DECIMAL(8,2). Si les pseudonymes n'existent pas, créez un mappage de type de données qui spécifie le type de source de données. Assurez-vous que l'encapsuleur de source de données a été créé avant d'exécuter l'instruction CREATE TYPE MAPPING. Pour mapper le type de données Oracle NUMBER(23,3) sur le type de données DB2 DECIMAL(8,2) pour les serveurs Oracle utilisant la version 8.0.3, exécutez l'instruction CREATE TYPE MAPPING, par exemple :

```
CREATE TYPE MAPPING ORA_DEC FROM SYSIBM.DECIMAL(8,2)
TO SERVER TYPE ORACLE VERSION 8.0.3 TYPE NUMBER(23,3)
```

ORA_DEC

Nom que vous attribuez au mappage de type. Ce nom ne peut pas dupliquer un nom de mappage de type de données qui existe déjà dans le catalogue.

FROM SYSIBM.DECIMAL(8,2)

Schéma DB2 local et type de données local. Si la longueur ou la précision et l'échelle ne sont pas spécifiés, ces valeurs sont déterminées à partir du type de données source.

TO SERVER TYPE ORACLE

Type de la source de données.

VERSION 8.0.3

Version du serveur de source de données. Vous devez indiquer la version. Vous pouvez également indiquer l'édition et la modification de l'édition, comme indiqué dans cet exemple.

TYPE NUMBER(23,3)

Type de données d'une source de données que vous mappez sur le type de données local. Les types de données définis par l'utilisateur ne sont pas autorisés.

La base de données fédérée définit le type de données DB2 DECIMAL(8,2) localement pour les colonnes Oracle sur des serveurs de la version 8.0.3. Les tables et vues Oracle sur des serveurs qui n'utilisent pas la version 8.0.3 empruntent plutôt le mappage de type de données par défaut. Lorsque vous créez des pseudonymes sur des tables et vues Oracle qui contiennent des colonnes NUMBER, le type de données Oracle NUMBER est mappé sur le type de données DB2 DECIMAL(8,2).

Création d'un mappage de type pour tous les objets de source de données sur un serveur – exemple

Dans cet exemple, le serveur est défini auprès de la base de données fédérée sous le nom de ORA2SERVER. Chaque table contient une colonne avec un type de données Oracle DATE.

Le type de données Oracle DATE contient le siècle, l'année, le mois, le jour, l'heure, la minute et la seconde. Le type de données Oracle DATE est mappé par défaut sur le type de données DB2 TIMESTAMP. Cependant, lorsqu'un objet fait l'objet d'une requête sur ce serveur, l'ensemble de résultats doit renvoyer uniquement les informations liées à l'heure (heure, minute et seconde).

Utilisez l'instruction ALTER NICKNAME pour modifier les types locaux de pseudonymes existants. Vous devez modifier chaque pseudonyme séparément pour modifier le type de données local en TIME.

Si les pseudonymes n'existent pas, créez un mappage de type de données qui spécifie le type de source de données.

Pour mapper le type de données Oracle DATE sur le type de données DB2 TIME pour ORA2SERVER, émettez l'instruction suivante :

```
CREATE TYPE MAPPING ORA2_DATE FROM SYSIBM.TIME  
TO SERVER ORA2SERVER TYPE DATE
```

ORA2_DATE

Nom que vous attribuez au mappage de type. Ce nom ne peut pas dupliquer un nom de mappage de type de données qui existe déjà dans le catalogue.

FROM SYSIBM.TIME

Schéma DB2 local et type de données local. Si la longueur ou la précision et l'échelle ne sont pas spécifiés, ces valeurs sont déterminées à partir du type de données source.

TO SERVER ORA2SERVER

Nom local du serveur de source de données.

TYPE DATE

Type de données d'une source de données que vous mappez sur le type de données local. Les types de données définis par l'utilisateur ne sont pas autorisés.

La base de données fédérée définit localement le type de données DB2 TIME pour les colonnes Oracle du type de données DATE.

Lorsque vous créez des pseudonymes sur des tables et vues Oracle qui contiennent des colonnes DATE, le type de données Oracle DATE est mappé sur le type de données DB2 DECIMAL(8,2).

Ce mappage de type de données n'a aucune conséquence sur les objets de source de données d'autres serveurs.

Transtypage entre types de données

Vous pouvez effectuer le transtypage d'un type de données depuis le type de données source vers le type de données cible.

Le transtypage entre types de données peut être implicite ou explicite.

- Le transtypage implicite est la conversion automatique de données d'un type de données vers un autre type de données selon un ensemble implicite de règles de conversion. Cette conversion automatique a lieu en soutien du typage faible.
- Le transtypage explicite soutient le typage fort. Le typage fort nécessite que les types de données coïncident. Vous devez convertir explicitement un type de données (ou les deux) en un type de données commun avant d'effectuer des comparaisons ou des affectations.

La prise en charge de la fédération pour le transtypage entre types de données permet des requêtes fédérées sur des pseudonymes afin d'accéder à des serveurs qui prennent en charge à la fois le typage fort et faible.

Les fonctions de transtypage ne peuvent pas être envoyées à des serveurs distants si aucune fonction équivalente n'existe sur le serveur distant. L'abus de transtypage peut entraîner des problèmes de performances.

Exemples : transtypage explicite

```
UPDATE nickname SET varcharcol = CAST(intcol AS varchar(10))  
  
SELECT REAL(varchar_col) FROM nickname1;  
  
SELECT VARCHAR(double_col) FROM nickname1;
```

Exemples : transtypage implicite

Exemple 1 :

```
UPDATE nickname SET varcharcol = intcol;
```

Dans cette opération d'affectation, l'instruction envoyée au serveur distant est équivalente à l'instruction suivante :

```
UPDATE nickname SET varcharcol = varchar(intcol);
```

Exemple 2 :

```
INSERT INTO nickname (varcharcol) SELECT intcol FROM nickname1;
```

Dans cette opération d'affectation, l'instruction envoyée au serveur distant est équivalente à l'instruction suivante :

```
INSERT INTO nickname (varcharcol) SELECT varchar(intcol) FROM nickname1;
```

Exemple 3 :

```
SELECT * SELECT nickname SELECT intcol = varcharcol;
```

Dans cette opération de comparaison, l'instruction envoyée au serveur distant est équivalente à l'instruction suivante :

```
SELECT * SELECT nickname SELECT intcol = CAST(varcharcol AS decfloat)
```

Prise en charge du type de données **TIMESTAMP**

Le type de données **TIMESTAMP** est paramétré pour contrôler la précision des secondes fractionnelles.

Pour la source de données DB2 pour Linux, UNIX et Windows, les horodatages distants effectuent un mappage vers **TIMESTAMP(*p*)**, où *p* représente la précision et définit le nombre de secondes fractionnelles. *p* est compris entre 0 et 12 inclus.

Pour les autres sources de données, les horodatages distants effectuent un mappage vers **TIMESTAMP** avec une précision par défaut de 6. Pour ces sources de données, vous pouvez profiter de la prise en charge **TIMESTAMP(*p*)** en utilisant des mappages similaires vers les échantillons fournis pour les mappages de type en aval.

Lorsqu'une colonne **TIMESTAMP** de pseudonymes a une précision inférieure à celle de la colonne de table distante correspondante et que la colonne de table distante contient des caractères numériques différents de zéro parmi les caractères numériques fractionnels supplémentaires, le résultat des prédicats utilisant cette colonne de pseudonymes peut être incertain. Le résultat des prédicats utilisant cette colonne de pseudonymes fonctionne correctement si le prédicat n'est pas renvoyé.

Modification d'un type local pour un objet de source de données (Centre de contrôle DB2)

Vous utilisez l'instruction **ALTER NICKNAME** au lieu de l'instruction **CREATE TYPE MAPPING** pour modifier un type de données local. Vous pouvez modifier le type de données à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit **SYSADM** ou **DBADM**
- privilège **ALTER** sur le pseudonyme spécifié dans l'instruction
- privilège **CONTROL** sur le pseudonyme spécifié dans l'instruction
- privilège **ALTERIN** sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne **DEFINER** de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Lorsque vous créez un pseudonyme, les types de données qui sont associés à l'objet source de données sont stockés dans la base de données fédérée. Pour certaines sources, l'encapsuleur indique les types de données. Pour d'autres sources, vous devez indiquer ces types en créant le pseudonyme.

Vous pouvez indiquer un type local pour une colonne d'un objet de source de données particulier.

Important : La modification du type de données local peut entraîner des erreurs ou une perte d'informations si vous remplacez le type de données local d'une colonne par un type très différent du type distant.

Procédure

Pour modifier le type de données local à partir du Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonyme**.
2. Cliquez avec le bouton droit de la souris sur le pseudonyme que vous souhaitez modifier puis cliquez sur **Modification**. Le bloc-notes Modification du pseudonyme s'ouvre.
3. Dans la page Pseudonyme, sélectionnez la colonne que vous souhaitez modifier puis cliquez sur **Modification**. La fenêtre Modification de la colonne s'ouvre.
4. Sélectionnez le type de données.
5. Cliquez sur **OK** pour changer le type de données et fermer la fenêtre.
6. Cliquez sur **OK** pour modifier le pseudonyme et fermer le bloc-notes.

Pour traiter le contenu d'une colonne locale dont le type de données caractères comme des données binaires, utilisez la clause FOR BIT DATA dans l'instruction ALTER NICKNAME. Lorsque vous utilisez cette clause pour modifier le type de données local d'une colonne, aucune conversion de pages de codes n'est effectuée lors de l'échange de données avec d'autres systèmes. Les comparaisons sont effectuées en binaire, quelle que soit la séquence de classement de la base de données distante.

Modification d'un type local d'objet de source de données – exemples

Cette rubrique fournit des exemples indiquant comment modifier les types de données d'un objet de source de données.

Exemple : Mappage de type de données numérique

Dans une table Oracle d'informations sur les employés, la colonne BONUS est définie avec le type de données NUMBER(32,3). Le type de données Oracle NUMBER(32,3) est mappé par défaut sur le type de données DB2 DOUBLE, un type de données numérique en virgule flottante en double précision. Une requête incluant la colonne BONUS peut renvoyer des valeurs semblables aux suivantes :

```
5.0000000000000E+002  
1.0000000000000E+003
```

La notation scientifique indique le nombre de décimales et le sens dans lequel le séparateur décimal doit être déplacé. Dans cet exemple, +002 signifie que le séparateur décimal doit être déplacé de deux positions vers la droite et +003 qu'il doit être déplacé de trois positions vers la droite.

Les requêtes qui incluent la colonne BONUS peuvent renvoyer des valeurs ressemblant à des montants en dollars. Vous remplacez le type de données DOUBLE de la définition locale de la colonne BONUS dans la table par le type de données DECIMAL. Utilisez une précision et une échelle qui reflètent le format des primes réelles. Par exemple, si la partie en dollars des primes ne doit pas dépasser six chiffres, mappez NUMBER(32,3) sur DECIMAL(8,2). Sous la contrainte de ce nouveau type local, les requêtes qui incluent la colonne BONUS renvoient des valeurs semblables aux suivantes :

```
500.00  
1000.00
```

Le pseudonyme de la table Oracle est ORASALES. Pour mapper la colonne BONUS de la table ORASALES sur le type de données DB2 DECIMAL (8,2), émettez l'instruction ALTER NICKNAME suivante :

```
ALTER NICKNAME ORASALES ALTER COLUMN BONUS  
LOCAL TYPE DECIMAL(8,2)
```

ORASALES

Pseudonyme que vous avez défini pour la table Oracle.

ALTER COLUMN BONUS

Nom de la colonne qui est défini localement dans la vue catalogue SYSCAT.COLUMNS de la base de données fédérée.

LOCAL TYPE DECIMAL(8,2)

Identifie le nouveau type local de la colonne.

Ce mappage ne s'applique qu'à la colonne BONUS de la table Oracle qui est identifiée par le pseudonyme ORASALES. Tous les autres objets de source de données Oracle qui incluent la colonne BONUS utilisent le mappage par défaut pour le type de données NUMBER Oracle.

Exemple : Mappage de données de type date

Le pseudonyme d'une table Oracle appelée SALES est ORASALES. La table SALES contient une colonne du type de données Oracle DATE. Le mappage par défaut du type de données Oracle DATE est sur le type de données DB2 TIMESTAMP. Cependant, vous souhaitez afficher uniquement la valeur de date lorsque vous extrayez des données de cette colonne. Vous pouvez modifier le pseudonyme de la table SALES pour remplacer le type local par le type de données DB2 DATE.

```
ALTER NICKNAME ORASALES ALTER COLUMN ORDER_DATE  
LOCAL TYPE DATE
```

Exemple : Mappage de type de données pour une source de données non relationnelle

Le pseudonyme d'un fichier structuré en tableaux appelé drugdata1.txt est DRUGDATA1. Le fichier drugdata1.txt contient une colonne qui dresse la liste de noms de produits pharmaceutiques. Le nom de colonne est DRUG. La colonne DRUG était définie à l'origine comme étant de type CHAR(20). La longueur de

cette colonne doit être modifiée en CHAR(30). Vous pouvez modifier le pseudonyme du fichier drugdata1.txt pour modifier le mappage à la longueur correcte.

```
ALTER NICKNAME DRUGDATA1 ALTER COLUMN DRUG  
LOCAL TYPE CHAR(30)
```

Modification d'un type local pour un objet de source de données (ligne de commande DB2)

Vous utilisez l'instruction ALTER NICKNAME au lieu de l'instruction CREATE TYPE MAPPING pour modifier un type de données local. Vous pouvez modifier le type de données à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Avant de commencer

L'ID autorisation qui émet l'instruction doit posséder au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- privilège ALTER sur le pseudonyme spécifié dans l'instruction
- privilège CONTROL sur le pseudonyme spécifié dans l'instruction
- privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
- créateur du pseudonyme enregistré dans la colonne DEFINER de la vue catalogue du pseudonyme

Restrictions

Voir «Restrictions pour les modifications de pseudonymes», à la page 34.

A propos de cette tâche

Lorsque vous créez un pseudonyme, les types de données qui sont associés à l'objet source de données sont stockés dans la base de données fédérée. Pour certaines sources, l'encapsuleur indique les types de données. Pour d'autres sources, vous devez indiquer ces types en créant le pseudonyme.

Vous pouvez indiquer un type local pour une colonne d'un objet de source de données particulier.

Important : La modification du type de données local peut entraîner des erreurs ou une perte d'informations si vous remplacez le type de données local d'une colonne par un type très différent du type distant.

Procédure

Pour modifier le type de données local à partir de l'invite de ligne de commande, utilisez l'instruction ALTER NICKNAME. Par exemple :

```
ALTER NICKNAME pseudonyme ALTER COLUMN nom_colonne  
LOCAL TYPE type_données
```

Pour traiter le contenu d'une colonne locale dont le type de données caractères comme des données binaires, utilisez la clause FOR BIT DATA dans l'instruction ALTER NICKNAME. Lorsque vous utilisez cette clause pour modifier le type de données local d'une colonne, aucune conversion de pages de codes n'est effectuée

lors de l'échange de données avec d'autres systèmes. Les comparaisons sont effectuées en binaire, quelle que soit la séquence de classement de la base de données distante.

Modification des types de données LONG en types de données VARCHAR

Pour autoriser les opérations d'insertion et de mise à jour sur les types de données LONG, vous pouvez transformer les données de type LONG en données de type VARCHAR.

Le tableau 5 répertorie les types de données LONG par source de données que vous pouvez modifier.

Tableau 5. Types de données LONG par source de données pouvant être transformés en types de données VARCHAR

Source de données	Type de données distant	Longueur	Type de données par défaut local	ALTER en VARCHAR
DRDA	LONG VARCHAR,	1 à 32672	CLOB	VARCHAR
	LONG VARCHAR FOR BIT DATA	1 à 32672	BLOB	VARCHAR FOR BIT DATA
Informix	BYTE	1 à 32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1 à 32672	CLOB	VARCHAR
Microsoft SQL Server	IMAGE	1 à 32672 variables hôte ; 1 à 8000 littéral	BLOB	VARCHAR FOR BIT DATA
	TEXT	1 à 32672 variables hôte ; 1 à 8000 littéral	CLOB	VARCHAR
Oracle NET8	LONG	1 à 32672 variables hôte ; 1 à 4000 littéral	CLOB	VARCHAR
	LONG RAW	1 à 32672 variables hôte ; 1 à 4000 littéral	BLOB	VARCHAR FOR BIT DATA
Sybase CTLIB	IMAGE	1 à 32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1 à 32672	CLOB	VARCHAR
Teradata	BYTE	32673 à 64000	BLOB	VARCHAR FOR BIT DATA(32672)
	CHAR	32673 à 64000	CLOB	VARCHAR(32672)
	VARBYTE	32673 à 64000	BLOB	VARCHAR FOR BIT DATA(32672)
	VARCHAR	32673 à 64000	CLOB	VARCHAR(32672)

Chapitre 4. Mappage de fonctions et de fonctions définies par l'utilisateur

Les mappages de fonctions associent les fonctions de serveur fédéré et les fonctions définies par l'utilisateur avec les fonctions existantes au niveau de la source de données.

Mappages de fonctions dans un système fédéré

IBM InfoSphere Federation Server fournit des mappages par défaut entre des fonctions de source de données existantes et des fonctions DB2 homologues.

Pour que le serveur fédéré utilise une fonction de source de données, un mappage est nécessaire entre une fonction DB2 ou un modèle de fonction et la fonction de source de données.

Les mappages de fonction par défaut figurent dans les modules d'encapsuleur.

Pour les sources de données non relationnelles, vous ne pouvez pas remplacer les mappages de fonctions existants ni en créer de nouveaux.

Principes de fonctionnement des mappages dans un système fédéré

Lors de la soumission de requêtes contenant une ou plusieurs fonctions au serveur fédéré, ce dernier vérifie les informations relatives aux mappages entre les fonctions DB2 et les fonctions de source de données.

Le serveur fédéré vérifie deux espaces contenant des informations sur les mappages :

- L'encapsuleur : l'encapsuleur de source de données contient les mappages de fonction par défaut.
- La vue catalogue SYSCAT.FUNCMAPPINGS : cette vue contient les entrées que vous avez créées et qui remplacent ou intensifient les mappages de fonction par défaut présents dans l'encapsuleur. Elle contient également de nouveaux mappages créés lorsqu'il n'y a pas mappage de fonction par défaut. Si plusieurs mappages peuvent être appliqués à une fonction, le dernier mappage créé est appliqué.

Les options de mappage de fonction contiennent des informations sur la fonction, ainsi que le coût potentiel du traitement d'une fonction au niveau de la source de données. Les options de mappage de fonction fournissent les informations suivantes :

- Nom de la fonction de source de données distante
- Estimation du nombre d'instructions traitées lors du premier et du dernier appel de la fonction de source de données.
- Estimation du nombre d'entrées-sorties lors du premier et du dernier appel de la fonction de source de données.
- Estimation du nombre d'instructions traitées par appel de fonction de source de données.

Lorsque vous créez un mappage de fonction, vous mappez depuis une fonction DB2 ou depuis un modèle de fonction vers une fonction homologue au niveau de la source de données. Lorsqu'aucune fonction DB2 homologue n'existe, ou que vous souhaitez forcer le serveur fédéré à utiliser la fonction de source de données, vous pouvez créer un modèle de fonction agissant comme homologue.

Pourquoi les mappages de fonction sont importants

Les mappages de fonction représentent l'une des entrées importantes pour l'analyse du transfert de l'exécution sur la source de données par l'optimiseur de requêtes.

En choisissant le meilleur plan d'accès aux requêtes, l'optimiseur de requêtes prend en compte les aptitudes de la source de données à exécuter un type d'opération ou de fonction SQL en particulier. Si la fonction n'a pas de mappage, elle n'est pas envoyée à la source de données pour traitement. Les fonctions et autres opérations qui peuvent être transférées à la source de données améliorent les performances.

Si la source de données a une fonction semblable à une fonction DB2, mais qu'elle renvoie des résultats légèrement différents, la création d'un mappage de fonction peut améliorer les performances. Par exemple, la fonction Informix STDEV (écart type) entraîne des résultats différents de la fonction DB2 STDDEV pour certains ensembles de données d'entrée. C'est pourquoi l'encapsuleur Informix n'a pas de mappage par défaut entre ces deux fonctions. Si vous ne vous souciez pas des différences entre les ensembles de résultats, vous pouvez améliorer les performances des requêtes qui accèdent aux sources de données Informix et utilisent la fonction DB2 STDDEV. En créant un mappage de fonction entre la fonction Informix STDEV et la fonction DB2 STDDEV, vous fournissez à l'optimiseur de requêtes la possibilité d'envoyer le traitement de cette fonction vers la source de données.

Quand créer les mappages de fonction

Lorsqu'aucun mappage de fonction par défaut n'est disponible pour une fonction de source de données, vous pouvez créer un mappage de fonction.

L'un des motifs expliquant l'indisponibilité d'un mappage de fonction est que la base de données fédérée n'a pas de fonction correspondant à la fonction de source de données.

Autre motif possible : la source de données a une fonction semblable à une fonction DB2, mais elle ne renvoie pas les mêmes résultats. Si la source de données renvoie des résultats légèrement différents ou des résultats différents pour certains ensembles de données d'entrée, les encapsuleurs ne mappent généralement pas vers ces fonctions. Toutefois, si les différences dans les ensembles de résultats ne vous importent pas, vous pouvez créer un mappage entre les fonctions. La création d'un mappage peut améliorer les performances.

Utilisez les mappages de fonction dans les cas suivants :

- Une nouvelle fonction intégrée devient disponible au niveau de la source de données
- Une nouvelle fonction définie par l'utilisateur devient disponible au niveau de la source de données
- Une fonction DB2 homologue n'existe pas
- Une fonction homologue existe mais elle renvoie des résultats légèrement différents qui ne vous sont pas utiles

Les paramètres des mappages de fonction sont stockés dans la vue catalogue SYSCAT.FUNCMAPPINGS.

Lorsque vous créez un mappage de fonction, il est possible que les valeurs renvoyées depuis une fonction évaluée au niveau de la source de données soient différentes des valeurs renvoyées depuis une fonction compatible évaluée au niveau de la base de données fédérée. IBM InfoSphere Federation Server utilise le mappage de fonction, mais il peut conduire à une erreur de syntaxe SQL ou à des résultats imprévus.

Fonctions définies par l'utilisateur dans les applications

Les développeurs d'application ont souvent besoin de créer leur propre suite de fonctions spécifiques à leur application ou à leur domaine. Dans cette optique, ils peuvent utiliser des fonctions scalaires définies par l'utilisateur.

Par exemple, supposons qu'un magasin de distribution puisse définir un type de données PRICE pour le suivi du coût des articles qu'il vend. Ce magasin peut également souhaiter définir une fonction SALES_TAX. Cette fonction prend alors une valeur de prix donnée sous la forme d'une entrée, elle calcule la taxe sur la valeur ajoutée applicable et renvoie ces données à l'application ou à l'utilisateur demandeur.

Ces fonctions sont utilisables sur tous les types de base de données, y compris les types d'objet LOB et les types distincts. Les fonctions définies par l'utilisateur permettent aux requêtes de contenir un calcul puissant ainsi que des prédicats de recherche pour filtrer les données non pertinentes proches de la source des données, ce qui réduit le temps de réponse. L'optimiseur SQL traite les fonctions définies par l'utilisateur exactement comme des fonctions intégrées telles que SUBSTR et LENGTH. Vous pouvez développer des applications à l'aide de divers environnements de langage d'application, tels que C, C++ et COBOL. Les applications peuvent partager un ensemble de fonctions SQL définies par l'utilisateur, même si elles sont développées à l'aide d'environnements de langage d'application différents.

Les fonctions définies par l'utilisateur peuvent manipuler les données et effectuer des actions. Par exemple, vous pouvez activer une fonction définie par l'utilisateur pour envoyer un courriel ou pour mettre à jour un fichier à plat.

Dans DB2, les fonctions définies par l'utilisateur peuvent inclure :

- Les fonctions que vous définissez de A à Z.
- Les fonctions du schéma SYSFUN. Les exemples incluent les fonctions mathématiques telles que SIN, COS et TAN ; les fonctions scientifiques telles que RADIANS, LOG10 et POWER ; et les fonctions générales telles que LEFT, DIFFERENCE et UCASE.

Exigences pour le mappage de fonctions définies par l'utilisateur

Avant de pouvoir appeler une fonction de source de données définie par l'utilisateur dans un système fédéré, la base de données fédérée doit associer la fonction de source de données à une spécification de fonction stockée dans le catalogue global, sur le serveur fédéré.

Deux conditions sont nécessaires à l'association d'une spécification de fonction à une fonction de source de données par la base de données fédérée :

- La base de données fédérée possède une fonction dont la signature correspond à celle de la fonction de source de données. Une *signature* comporte un nom de fonction et des paramètres d'entrée de fonction. Les signatures *correspondent* lorsque les deux conditions suivantes sont vraies :
 - Elles contiennent les mêmes noms et le même nombre de paramètres
 - Le type de données de chaque paramètre d'une signature est identique (ou peut être converti sur) au type de données du paramètre correspondant de l'autre signature.
- Si la base de données fédérée ne comporte pas de fonction ayant la signature requise, vous pouvez définir un modèle de fonction contenant cette signature. Ensuite, vous mappez le modèle de fonction vers la fonction de source de données à appeler.

Les paramètres des mappages de fonction sont stockés dans la vue catalogue SYSCAT.FUNCMAPPINGS.

Création de mappages de fonction

L'instruction CREATE FUNCTION MAPPING permet de spécifier des mappages de fonction alternatifs qui remplacent les mappages de fonction par défaut.

Lors de la création de mappages de fonction alternatifs, les entrées apparaissent dans la vue catalogue SYSCAT.FUNCMAPPINGS.

Vous pouvez également utiliser l'instruction CREATE FUNCTION MAPPING pour spécifier des options de mappage de fonction. Lorsque vous en indiquez, les informations apparaissent dans la vue catalogue SYSCAT.FUNCMAPOPTIONS.

L'instruction CREATE FUNCTION MAPPING permet de :

- Créer un mappage de fonction pour toutes les sources de données d'un type spécifique. Par exemple, toutes les sources de données Informix.
- Créer un mappage de fonction pour toutes les sources de données d'un type et d'une version spécifiques. Par exemple, toutes les sources de données Informix 9.
- Créer un mappage de fonction pour un serveur spécifique.
- Fournir des informations statistiques sur les mappages de fonction à l'optimiseur
- Désactiver un mappage de fonction par défaut ou un mappage de fonction que vous avez défini.

Vous pouvez émettre l'instruction CREATE FUNCTION MAPPING dans l'interpréteur de commandes. Vous pouvez également incorporer l'instruction CREATE FUNCTION MAPPING à un programme d'application. Le Centre de contrôle DB2 ne prend en charge ni la création, ni la modification de mappages de fonction.

Spécification de noms de fonction dans l'instruction CREATE FUNCTION MAPPING

Les valeurs saisies dans l'instruction CREATE FUNCTION MAPPING dépendent selon que les fonctions que vous mappez ensemble portent, ou non, des noms identiques.

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

Mappage des fonctions portant un nom identique

Vous pouvez créer un mappage entre deux fonctions (ou entre un modèle de fonction DB2 et une fonction de source de données) portant le même nom.

Procédure

Pour mapper deux fonctions portant le même nom, émettez l'instruction CREATE FUNCTION MAPPING.

Exemple : supposons que vous vouliez mapper une fonction définie par l'utilisateur intitulée MYFUN sur une source de données Informix avec la fonction DB2 définie par l'utilisateur, intitulée TINA.MYFUN. Le serveur de source de données Informix s'appelle INFORMIX2. L'instruction suivante mappe la fonction :

```
CREATE FUNCTION MAPPING FOR TINA.MYFUN(SYSTEM.INTEGER) SERVER INFORMIX2
```

Mappage de fonctions portant des noms différents

Vous pouvez créer un mappage entre deux fonctions (ou entre un modèle de fonction DB2 et une fonction de source de données) portant des noms différents.

Pour créer un mappage entre deux fonctions portant des noms différents, émettez l'instruction CREATE FUNCTION MAPPING :

1. Attribuez le nom de la fonction DB2 ou du modèle de fonction au paramètre nom_de_fonction.
2. Spécifiez une option de mappage de fonction intitulée REMOTE_NAME et attribuez à cette option le nom de la fonction de source de données. REMOTE_NAME doit comporter moins de 255 caractères.

Exemple : supposons que vous vouliez mapper une fonction définie par l'utilisateur intitulée UPPERCASE sur une source de données Oracle avec la fonction DB2 UCASE(CHAR). Le nom du serveur de source de données Oracle est ORACLE2. Vous choisissez d'appeler ce mappage de fonction ORACLE_UPPER. La syntaxe apparaît comme suit :

```
CREATE FUNCTION MAPPING ORACLE_UPPER FOR SYSFUN.UCASE(CHAR)
  SERVER ORACLE2 OPTIONS
  (REMOTE_NAME 'UPPERCASE')
```

Création d'un mappage de fonction pour un type de source de données spécifique

Vous pouvez créer un mappage vers une fonction pour toutes les sources de données d'un type en particulier.

Avant de commencer

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

Restrictions

Vous ne pouvez ni remplacer les mappages de fonction existants ni en créer de nouveaux pour les sources de données non relationnelles.

Procédure

Pour mapper un modèle de fonction DB2 vers une fonction de source de données, utilisez l'instruction CREATE FUNCTION MAPPING.

Exemple : mappage d'un modèle de fonction DB2 vers une fonction Oracle définie par l'utilisateur pour toutes les sources de données Oracle

```
CREATE FUNCTION MAPPING MY_ORACLE_FUN1
  FOR NOVA.STATS ( DOUBLE, DOUBLE )
  SERVER TYPE ORACLE
  OPTIONS (REMOTE_NAME 'STAR.STATISTICS')
```

Le modèle s'intitule STATS et appartient à un schéma appelé NOVA. Le nom de la fonction Oracle définie par l'utilisateur est STATISTICS et elle appartient au schéma STAR.

Création d'un mappage de fonction pour une version et un type de source de données spécifiques

Vous pouvez créer un mappage vers une fonction pour toutes les sources de données qui utilisent une version spécifique du type de source de données.

Avant de commencer

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

Restrictions

Vous ne pouvez ni remplacer les mappages de fonction existants ni en créer de nouveaux pour les sources de données non relationnelles.

Procédure

Pour créer un mappage pour une version et un type de source de données spécifiques, utilisez l'instruction CREATE FUNCTION MAPPING.

Exemple : mappage d'un modèle de fonction DB2 vers une fonction Sybase définie par l'utilisateur pour toutes les sources de données Sybase utilisant la version 15

Le modèle s'intitule SYB_STATS et il appartient au schéma EARTH. Le nom de la fonction Sybase définie par l'utilisateur est STATISTICS et elle appartient au schéma MOON. L'instruction CREATE FUNCTION MAPPING apparaît comme suit :

```
CREATE FUNCTION MAPPING SYBASE_STATS
  FOR EARTH.SYB_STATS ( DOUBLE, DOUBLE )
  SERVER TYPE SYBASE VERSION 15
  OPTIONS (REMOTE_NAME 'MOON.STATISTICS')
```

Création d'un mappage de fonction pour tous les objets de source de données sur un serveur spécifique

Vous pouvez créer un mappage vers une fonction utilisé par toutes les objets de sources de données sur un serveur distant en particulier.

Avant de commencer

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

Restrictions

Vous ne pouvez ni remplacer les mappages de fonction existants ni en créer de nouveaux pour les sources de données non relationnelles.

Procédure

Pour créer un mappage de fonction pour tous les objets de source de données sur un serveur spécifique, utilisez l'instruction CREATE FUNCTION MAPPING.

Exemple : mappage d'un modèle de fonction intitulé **BONUS** vers une fonction définie par l'utilisateur appelée **BONUS**

Supposons que vous vouliez que le mappage s'applique uniquement à un serveur de source de données Oracle appelé ORA_SALES. Puisque les noms des fonctions sont identiques, vous n'êtes pas tenu de préciser l'option de mappage de fonction REMOTE_NAME.

```
CREATE FUNCTION MAPPING BONUS_CALC FOR BONUS()  
SERVER ORA_SALES
```

Modèles de fonctions

Le serveur fédéré reconnaît une fonction de source de données en cas de mappage entre la fonction de source de données et une fonction DB2 homologue au niveau de la base de données fédérée.

Vous pouvez créer un modèle de fonction agissant en tant que fonction DB2 homologue lorsque aucun homologue n'existe.

Un *modèle de fonction* est une fonction DB2 créée dans l'optique de forcer le serveur fédéré à appeler une fonction de source de données. Toutefois, contrairement à une fonction classique, un modèle de fonction n'a pas de code exécutable. Lorsque le serveur fédéré reçoit des requêtes spécifiant le modèle de fonction, il appelle la fonction de source de données.

Le modèle de fonction est créé avec l'instruction CREATE FUNCTION à l'aide du paramètre AS TEMPLATE.

Une fois le modèle de fonction créé, créez ensuite le mappage de fonction entre le modèle et la fonction de source de données. L'instruction CREATE FUNCTION MAPPING permet de créer un mappage de fonction.

Création de modèles de fonction

Le serveur fédéré reconnaît une fonction de source de données en cas de mappage entre la fonction de source de données et une fonction homologue au niveau de la base de données fédérée. Vous pouvez créer un modèle de fonction agissant en tant qu'homologue lorsque aucun homologue n'existe.

Avant de commencer

L'ID autorisation de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- Le droit IMPICIT_SCHEMA sur les bases de données, si le nom de schéma implicite ou explicite de la fonction n'existe pas
- Le privilège CREATEIN sur le schéma, si le nom de schéma de la fonction existe

Restrictions

Si la fonction de source de données comporte des paramètres d'entrée :

- La fonction DB2 homologue doit comporter le même nombre de paramètres d'entrée que la fonction de source de données.
- Les types de données des paramètres d'entrée pour la fonction DB2 homologue doivent être compatibles avec les types de données correspondants des paramètres d'entrée pour la fonction de source de données. Le type de données ne peut pas être LONG VARCHAR, LONG VARGRAPHIC ou un type défini par l'utilisateur.

Si la fonction de source de données ne comporte pas de paramètres d'entrée, la fonction DB2 homologue ne peut pas en avoir.

Procédure

Pour créer un modèle de fonction, procédez comme suit :

1. Utiliser l'instruction CREATE FUNCTION avec le paramètre AS TEMPLATE.

Par exemple :

```
CREATE FUNCTION BONUS ()  
  RETURNS DECIMAL(8,2)  
  AS TEMPLATE  
  DETERMINISTIC  
  NO EXTERNAL ACTION
```

BONUS ()

Nom attribué au modèle de fonction.

RETURNS *DECIMAL(8,2)*

Type de données de la sortie.

AS TEMPLATE

Indique qu'il s'agit d'un modèle de fonction et pas d'une fonction.

DETERMINISTIC

Indique que la fonction renvoie toujours les mêmes résultats pour un ensemble de valeurs d'arguments.

NO EXTERNAL ACTION

Indique que la fonction n'a pas d'impact externe sur les objets non gérés par le gestionnaire de base de données.

Vous devez spécifier les clauses DETERMINISTIC et NO EXTERNAL ACTION selon que la fonction est elle-même déterministe, ou non, et si cela provoque une action externe. Sinon, les restrictions seront imposées aux opérations SQL prises en charge avec ce modèle de fonction.

2. Une fois le modèle de fonction créé, vous devez ensuite créer le mappage de fonction entre le modèle et la fonction de source de données. Un mappage de fonction est créé à l'aide de l'instruction CREATE FUNCTION MAPPING. Par exemple :

```
CREATE FUNCTION MAPPING MY_INFORMIX_FUN FOR BONUS()  
SERVER TYPE INFORMIX OPTIONS (REMOTE_NAME 'BONUS()')
```

MY_INFORMIX_FUN

Nom attribué au mappage de fonction. Le nom ne peut pas être identique à un nom de mappage de fonction déjà décrit dans le catalogue global de la base de données fédérée. Il doit être unique.

FOR BONUS()

Nom du modèle de fonction DB2 local. Les paramètres d'entrée du type de données sont placés entre parenthèses.

SERVER TYPE INFORMIX

Identifie le type de source de données contenant la fonction vers laquelle vous voulez mapper.

OPTIONS (REMOTE_NAME 'BONUS()')

Option identifiant le nom de la fonction de source de données distante que vous mappez vers le modèle de fonction DB2 local.

Désactivation d'un mappage de fonction par défaut

Les mappages de fonction par défaut ne peuvent pas être supprimés. Vous pouvez toutefois les rendre inutilisables en les désactivant.

Avant de commencer

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

Procédure

Pour désactiver un mappage de fonction par défaut, l'instruction CREATE FUNCTION MAPPING indique le nom de la fonction DB2 et définit l'option DISABLE sur 'Y'.

Exemple : désactivation d'un mappage de fonction par défaut entre la fonction DB2 SIN et une fonction similaire sur les sources de données Oracle

Lors du traitement d'une requête interrogeant les données Oracle et faisant référence à la fonction SIN, l'une des fonctions peut être appelée. La fonction appelée dépend de la fonction évaluée par l'optimiseur de requêtes comme nécessitant un temps système moindre.

Pour vous assurer que la fonction DB2 SIN est appelée et que la fonction Oracle SIN ne l'est pas, vous devez désactiver le mappage de fonction par défaut. Utilisez la syntaxe suivante :

```
CREATE FUNCTION MAPPING FOR SYSFUN.SIN(INT)  
TYPE ORACLE OPTIONS (DISABLE 'Y')
```

Suppression d'un mappage de fonction

Lorsque vous n'avez plus besoin d'un mappage de fonction que vous avez créé, vous pouvez le supprimer en exécutant l'instruction DROP.

Avant de commencer

L'identificateur utilisateur de l'instruction doit disposer du droit SYSADM ou DBADM.

A propos de cette tâche

Si vous supprimez un mappage de fonction que vous avez créé pour le remplacer par un mappage de fonction par défaut, ce dernier est utilisé.

Les mappages de fonction sont répertoriés dans la vue catalogue SYSCAT.FUNCMAPPINGS.

Procédure

Pour supprimer un mappage de fonction que vous avez créé, utilisez l'instruction DROP :

Exemple : annulation d'un mappage de fonction intitulé BONUS_CALC

```
DROP FUNCTION MAPPING BONUS_CALC
```

Chapitre 5. Création de spécifications d'index

Pour les sources de données relationnelles, la création de spécifications d'index permet de stocker des informations sur les colonnes des index distants dans le catalogue global et de garantir des performances optimales au niveau des requêtes de recherche.

Spécifications d'index dans un système fédéré

Dans un système fédéré, l'instruction CREATE INDEX utilisée avec un pseudonyme permet de stocker dans le catalogue global des informations sur la disponibilité d'un index sur l'objet distant. L'optimiseur de requêtes utilise ces informations pour optimiser les requêtes.

Lors de l'émission d'une instruction CREATE INDEX :

- Si un pseudonyme est créé pour une table, l'instruction CREATE INDEX collecte les informations relatives à l'index créé sur la table distante.
- Si un pseudonyme est créé pour une vue, l'instruction CREATE INDEX fait référence au pseudonyme de la vue et contient des informations relatives à l'index sur la table sous-jacente à la vue.

La spécification d'index informe le serveur fédéré sur les colonnes qui composent l'index distant, et sur leurs propriétés d'unicité. Elle ne l'informe pas sur les propriétés statistiques de l'index, telles que le nombre de valeurs uniques de la clé d'index.

Vous n'êtes pas tenu de fournir des spécifications d'index si l'index distant était en place lors de la création du pseudonyme.

Un serveur fédéré ne crée pas de spécification d'index lorsque vous créez un pseudonyme pour :

- Une table n'ayant pas d'index
- Une vue qui ne dispose généralement pas d'informations d'index stockées dans le catalogue distant
- Un objet de source de données n'ayant pas de catalogue distant à partir duquel le serveur fédéré peut obtenir des informations d'index

Supposons qu'une table acquiert un nouvel index, en plus de ceux qu'elle avait lorsque le pseudonyme a été créé. Puisque les informations d'index sont fournies au catalogue global au moment où le pseudonyme est créé, le serveur fédéré n'a pas connaissance du nouvel index. De même, lorsqu'un pseudonyme est créé pour une vue, le serveur fédéré n'a pas connaissance de la table sous-jacente (et de ses index) à partir de laquelle la vue a été générée. Dans ces circonstances, vous pouvez fournir au catalogue global les informations d'index nécessaires. Vous pouvez créer une spécification d'index pour les tables n'ayant pas d'index. La spécification d'index indique à l'optimiseur de requêtes la ou les colonnes dans lesquelles la recherche doit être effectuée pour trouver rapidement des données.

Utilisez les spécifications d'index avec les sources de données relationnelles. La création d'une spécification d'index pour une source de données non relationnelles n'améliore pas les performances.

Création des spécifications d'index pour les objets de source de données

Lorsqu'un pseudonyme est créé pour une table de source de données, le serveur fédéré fournit au catalogue global des informations sur chaque index de la table de source de données. L'optimiseur utilise ces informations pour accélérer le traitement des requêtes distribuées. Ces informations constituent un ensemble de métadonnées et sont appelées *spécification d'index*.

Avant de commencer

L'identificateur utilisateur de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- Le privilège CONTROL ou INDEX sur l'objet. Ainsi que le droit IMPLICIT_SCHEMA sur la base de données (si le nom de schéma implicite ou explicite de l'index n'existe pas), ou le privilège CREATEIN sur le schéma (si le nom de schéma de l'index fait référence à un schéma existant).

Restrictions

La création d'une spécification d'index sur un pseudonyme est soumise à certaines restrictions.

- Si l'option de liaison DYNAMICRULES BIND s'applique, l'instruction ne peut pas être préparée de manière dynamique. En outre, vous ne pouvez pas utiliser les paramètres INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS et ALLOW REVERSE SCANS dans l'instruction CREATE INDEX.
- L'option UNIQUE doit être spécifiée uniquement si les données de la clé d'index contiennent des valeurs uniques pour chaque ligne de la table de source de données. L'unicité n'est pas vérifiée.
- La somme des longueurs mémorisées des colonnes spécifiées ne doit pas être supérieure à 1024.
- Aucune colonne LOB ou colonne de type distinct basée sur une LOB ne peut être utilisée dans le cadre d'un index. Cette restriction s'applique même si l'attribut de longueur de la colonne s'intègre dans la limite fixée à 1024 octets.

A propos de cette tâche

Le serveur fédéré ne crée pas de spécification d'index dans les situations suivantes :

- Un pseudonyme est créé pour une table sans index.
- Un pseudonyme est créé pour un objet de source de données ne contenant pas d'index tel qu'une vue ou un synonyme Informix.
- Un pseudonyme est créé pour un objet non relationnel, par exemple un fichier structuré en table, un tableur Excel ou un fichier XML catégorisé.
- L'index distant est sur une colonne LOB ou XML.
- L'index distant contient une longueur de clé totale supérieure à 1024 octets.
- Le nombre maximal de composants de clés est supérieur à 16.

Dans ces situations, le serveur fédéré ne stocke pas de spécification d'index pour les objets de source de données. Toutefois, pour les deux premiers éléments de la liste précédente, vous pouvez fournir les informations d'index nécessaires au catalogue global. L'instruction CREATE INDEX permet de spécifier les informations d'index.

Procédure

Pour créer un index, vous pouvez incorporer l'instruction CREATE INDEX dans un programme d'application ou émettre l'instruction en tant qu'instruction SQL dynamique à partir du Centre de contrôle ou de la ligne de commande.

Lorsqu'elle est utilisée avec des pseudonymes, l'instruction CREATE INDEX crée une spécification d'index dans le catalogue global fédéré. Elle ne crée pas d'index dans la table de source de données.

Pour créer une spécification d'index, utilisez la syntaxe suivante :

```
CREATE INDEX nom_index ON pseudonyme  
(nom_de_colonne) SPECIFICATION ONLY  
CREATE UNIQUE INDEX nom_index ON pseudonyme  
(nom_de_colonne DESC) SPECIFICATION ONLY
```

Pour une spécification d'index, *nom_de_colonne* correspond au nom par lequel le serveur fédéré fait référence à une colonne d'une table de source de données.

Création de spécifications d'index sur les tables acquérant de nouveaux index

Lorsqu'une table acquiert un nouvel index, vous devez créer une spécification d'index sur le pseudonyme correspondant à la table.

Avant de commencer

L'identificateur utilisateur de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- Le privilège CONTROL ou INDEX sur l'objet. Ainsi que le droit IMPLICIT_SCHEMA sur la base de données (si le nom de schéma implicite ou explicite de l'index n'existe pas), ou le privilège CREATEIN sur le schéma (si le nom de schéma de l'index fait référence à un schéma existant).

Restrictions

La création d'un index sur un pseudonyme est soumise à certaines restrictions.

- Si l'option de liaison DYNAMICRULES BIND s'applique, l'instruction ne peut pas être préparée de manière dynamique. En outre, vous ne pouvez pas utiliser les paramètres INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS et ALLOW REVERSE SCANS dans l'instruction CREATE INDEX.
- UNIQUE doit être spécifié uniquement si les données de la clé d'index contiennent des valeurs uniques pour chaque ligne de la table de source de données. L'unicité n'est pas vérifiée.
- La somme des longueurs mémorisées des colonnes spécifiées ne doit pas être supérieure à 1024.

- Aucune colonne LOB ou colonne de type distinct basée sur une LOB ne peut être utilisée dans le cadre d'un index. Cette restriction s'applique même si l'attribut de longueur de la colonne s'intègre dans la limite fixée à 1024 octets.

A propos de cette tâche

Plusieurs situations entraînent l'acquisition d'un nouvel index par une table :

- Vous créez un pseudonyme pour une table sans index, mais qui en acquiert un ultérieurement
- Vous créez un pseudonyme pour une table ayant un index, et qui en acquiert un autre ultérieurement

Dans ces situations, vous devez créer une spécification d'index pour la table de sorte que le compilateur SQL puisse utiliser ces informations lors du traitement des requêtes faisant référence à la table.

Procédure

Les exemples suivants décrivent la procédure de création d'une spécification d'index pour un pseudonyme correspondant à une table acquérant un index.

Exemple : une table n'a pas d'index, et en acquiert un ultérieurement

Supposons que vous créez le pseudonyme *EMPLOYEE* pour une table de source de données intitulée *CURRENT_EMP*, qui n'a pas d'index. Parfois, une fois le pseudonyme créé, un index est défini sur *CURRENT_EMP* à l'aide des colonnes *WORKDEPT* et *JOB* pour la clé d'index.

Pour créer une spécification d'index décrivant l'index, présentez la syntaxe comme suit :

```
CREATE UNIQUE INDEX JOB_BY_DEPT ON EMPLOYEE  
(WORKDEPT, JOB) SPECIFICATION ONLY
```

où *JOB_BY_DEPT* correspond au nom d'index.

Exemple : une table acquiert un nouvel index

Supposons que vous créez le pseudonyme *JP_SALES* pour une table intitulée *JAPAN_SALES*. Un nouvel index est ensuite ajouté à la table, en plus de ceux qu'elle possédait déjà lors de la création du pseudonyme. Le nouvel index utilise la colonne *MARKUP* pour la clé d'index.

Pour créer une spécification d'index décrivant l'index, présentez la syntaxe comme suit :

```
CREATE UNIQUE INDEX JP_MARKUP ON JP_SALES (MARKUP) SPECIFICATION ONLY
```

où *JP_MARKUP* correspond au nom d'index.

Création de spécifications d'index sur les vues

Lorsqu'un pseudonyme est créé pour une vue, le serveur fédéré n'a pas connaissance de la table sous-jacente (et de ses index) à partir de laquelle la vue a été générée. Créez une spécification d'index pour la vue, de sorte que le compilateur SQL puisse utiliser ces informations lors du traitement des requêtes faisant référence à la vue.

Avant de commencer

L'identificateur utilisateur de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- Le privilège CONTROL ou INDEX sur l'objet. Ainsi que le droit IMPLICIT_SCHEMA sur la base de données (si le nom de schéma implicite ou explicite de l'index n'existe pas), ou le privilège CREATEIN sur le schéma (si le nom de schéma de l'index fait référence à un schéma existant).

Restrictions

La création d'un index sur un pseudonyme est soumise à certaines restrictions.

- Si l'option de liaison DYNAMICRULES BIND s'applique, l'instruction ne peut pas être préparée de manière dynamique. En outre, vous ne pouvez pas utiliser les paramètres INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS et ALLOW REVERSE SCANS dans l'instruction CREATE INDEX.
- UNIQUE doit être spécifié uniquement si les données de la clé d'index contiennent des valeurs uniques pour chaque ligne de la table de source de données. L'unicité n'est pas vérifiée.
- La somme des longueurs mémorisées des colonnes spécifiées ne doit pas être supérieure à 1024.
- Aucune colonne LOB ou colonne de type distinct basée sur une LOB ne peut être utilisée dans le cadre d'un index. Cette restriction s'applique même si l'attribut de longueur de la colonne s'intègre dans la limite fixée à 1024 octets.

Procédure

Pour créer une spécification d'index pour une vue, procédez comme suit :

- Assurez-vous que la ou les colonnes sur lesquelles l'index de la table est basé font partie de la vue.
- Si vous voulez créer des spécifications d'index pour tous les index dans une table sous-jacente, chaque spécification d'index doit être créée séparément.

Exemple : création d'une spécification d'index décrivant l'index REGION

Supposons que vous créez le pseudonyme *JP_SALES2007* pour une vue intitulée JAPAN_SALES2007. La table sous-jacente de cette vue est la table JAPAN_SALES qui contient plusieurs index : REGION, AMOUNT, SALES_REP. L'instruction CREATE INDEX créée fait référence au pseudonyme de la vue et contient des informations sur l'index de la table sous-jacente de la vue.

```
CREATE UNIQUE INDEX JP_2007_REGION ON JP_SALES2007  
(REGION) SPECIFICATION ONLY
```

où *JP_2007_REGION* correspond au nom d'index et *JP_SALES2007* correspond au pseudonyme de la vue JAPAN_SALES2007.

Création de spécifications d'index sur les synonymes Informix

Cette rubrique décrit la mesure appliquée par le serveur fédéré pour les synonymes Informix basés sur une table ou une vue :

Avant de commencer

L'identificateur utilisateur de l'instruction doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- Le privilège CONTROL ou INDEX sur l'objet. Ainsi que le droit IMPLICIT_SCHEMA sur la base de données (si le nom de schéma implicite ou explicite de l'index n'existe pas), ou le privilège CREATEIN sur le schéma (si le nom de schéma de l'index fait référence à un schéma existant).

Restrictions

La création d'un index sur un pseudonyme est soumise à certaines restrictions.

- Si l'option de liaison DYNAMICRULES BIND s'applique, l'instruction ne peut pas être préparée de manière dynamique. En outre, vous ne pouvez pas utiliser les paramètres INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS et ALLOW REVERSE SCANS dans l'instruction CREATE INDEX.
- UNIQUE doit être spécifié uniquement si les données de la clé d'index contiennent des valeurs uniques pour chaque ligne de la table de source de données. L'unicité n'est pas vérifiée.
- La somme des longueurs mémorisées des colonnes spécifiées ne doit pas être supérieure à 1024.
- Aucune colonne LOB ou colonne de type distinct basée sur une LOB ne peut être utilisée dans le cadre d'un index. Cette restriction s'applique même si l'attribut de longueur de la colonne s'intègre dans la limite fixée à 1024 octets.

A propos de cette tâche

Dans Informix, vous pouvez créer un synonyme pour une table ou une vue. Puisque le serveur fédéré vous permet de créer des pseudonymes pour les synonymes Informix, la mesure prise par le serveur fédéré dépend de l'élément sur lequel le synonyme est basé : une table ou une vue.

- Supposons qu'un pseudonyme soit créé pour un synonyme, et que ce synonyme soit basé sur une table Informix. Si le serveur fédéré détermine que la table à laquelle le synonyme fait référence possède un index, une spécification d'index est alors créée pour le synonyme. Si la table à laquelle le synonyme fait référence ne possède pas d'index, aucune spécification d'index n'est créée pour le synonyme. Vous pouvez toutefois créer une spécification d'index manuellement, à l'aide de l'instruction CREATE INDEX.
- Supposons qu'un pseudonyme soit créé pour un synonyme, et que ce synonyme soit basé sur une vue Informix. Le serveur fédéré ne peut pas déterminer sur quelle(s) table(s) sous-jacente(s) la vue est basée. Ainsi, aucune spécification d'index n'est créée pour le synonyme. Vous pouvez toutefois créer manuellement une spécification d'index à l'aide de l'instruction CREATE INDEX.

Procédure

Les exemples suivants décrivent la procédure de création d'une spécification d'index sur un pseudonyme correspondant à un synonyme Informix.

Exemple : un pseudonyme est créé sur un synonyme Informix basé sur une table

Lorsque le synonyme est basé sur une table Informix ne contenant pas d'index, vous pouvez créer une spécification d'index pour le synonyme, afin d'informer l'optimiseur de la ou des colonnes à utiliser pour rechercher rapidement des données. L'instruction créée spécifie le pseudonyme du synonyme, et vous fournissez les informations relatives à la ou aux colonnes dans la table sur laquelle le synonyme est basé.

Dans cet exemple, vous créez le pseudonyme *CONTRACTS* pour un synonyme appelé *SALES_CONTRACTS*. La table sur laquelle ce synonyme est basé s'intitule *SALES2006_TABLE* et contient plusieurs index : *REGION*, *AMOUNT*, *SALES_REP*. L'instruction *CREATE INDEX* créée fait référence au pseudonyme du synonyme et contient des informations sur l'index de la table sous-jacente du synonyme.

Pour créer une spécification d'index qui décrit l'index *REGION*, présentez la syntaxe comme suit :

```
CREATE UNIQUE INDEX NORTHWEST_2006_REGION ON CONTRACTS (REGION) SPECIFICATION ONLY
```

où *NORTHWEST_2006_REGION* correspond au nom d'index et *CONTRACTS* correspond au pseudonyme du synonyme *SALES_CONTRACTS*.

Exemple : un pseudonyme est créé sur un synonyme Informix basé sur une vue

Vous créez le pseudonyme *JP_SALES2007* pour un synonyme basé sur une vue intitulée *JAPAN_SALES2007*. La table sous-jacente de cette vue est la table *JAPAN_SALES* qui contient plusieurs index : *REGION*, *AMOUNT*, *SALES_REP*. L'instruction *CREATE INDEX* créée fait référence au pseudonyme du synonyme et contient des informations sur l'index de la table sous-jacente de la vue.

Lors de la création d'une spécification d'index pour un synonyme basé sur une vue, assurez-vous que la ou les colonnes sur lesquelles l'index de la table est basé font partie de la vue. Si vous voulez créer des spécifications d'index pour tous les index dans une table sous-jacente, chaque spécification d'index doit être créée séparément.

Pour créer une spécification d'index qui décrit l'index *REGION*, présentez la syntaxe comme suit :

```
CREATE UNIQUE INDEX JP_2007_REGION ON JP_SALES2007 (REGION) SPECIFICATION ONLY
```

où *JP_2007_REGION* correspond au nom d'index et *JP_SALES2007* correspond au pseudonyme de la vue *JAPAN_SALES2007*.

Chapitre 6. Développement de procédures fédérées

Les procédures fédérées vous permettent d'appeler des procédures dans une source de données comme si la procédure éloignée était une procédure locale.

Procédures fédérées

Une *procédure fédérée* est un objet de base de données fédérée qui référence une procédure sur une source de données.

Les procédures fédérées ne sont pas des noms alternatifs pour les procédures de sources de données, contrairement aux alias qui sont des noms alternatifs. Une procédure fédérée est définie au niveau de la base de données fédérée, mais elle appelle une procédure de source de données lorsque la procédure fédérée est invoquée. Etant donné que la procédure fédérée est un objet de base de données fédérée, les utilisateurs et les applications client peuvent invoquer la logique de procédure de source de données en appelant une procédure fédérée. Les résultats de la procédure de source de données, tels que les paramètres de sortie, sont renvoyés par la procédure fédérée. L'utilisation d'une procédure fédérée rend l'emplacement de la procédure de source de données transparent pour les utilisateurs et les applications client. Utilisez le nom de la procédure fédérée pour appeler la procédure de source de données.

Une procédure fédérée est à une procédure éloignée ce qu'un pseudonyme est pour une table éloignée. Les pseudonymes et les procédures fédérées sont des objets de la base de données fédérée. Un pseudonyme est un objet qui référence un objet, comme une table ou une vue, sur la source de données. Il vous permet d'interroger un objet de source de données. Une procédure fédérée vous permet d'appeler une procédure de source de données.

Utilisez l'instruction CREATE PROCEDURE (dérivée) pour enregistrer une procédure fédérée et utilisez une instruction CALL pour appeler une procédure. Vous pouvez intégrer l'instruction CREATE PROCEDURE (dérivée) à un programme d'application ou émettre l'instruction à l'aide d'instructions SQL dynamiques.

Paramètres et types de données

La fédération utilise trois types de paramètres : IN, OUT et INOUT. Lors de la création d'une procédure fédérée, les mappages de types de données aval par défaut permettent de mapper les types de données des paramètres de la procédure de source de données par rapport aux types de données fédérées. Ce mappage contient des types de données pour les paramètres d'entrée et de sortie et des types de données pour les colonnes de l'ensemble de résultats. Vous pouvez utiliser l'instruction CREATE TYPE MAPPING pour remplacer le mappage de types de données par défaut des paramètres de source de données. Toutefois, les mappages de types de données de l'ensemble de résultats ne sont pas affectés par ceux définis par l'utilisateur. Vous pouvez utiliser tous les types de données pris en charge dans les colonnes de pseudonymes, sauf les types de données LOB. Les procédures fédérées ne prennent pas en charge des types de données complexes.

Mappages utilisateur et autorisation

Pour appeler une procédure fédérée, vous devez disposer des autorisations adéquates pour la procédure fédérée et la procédure de source de données. Lorsque vous appelez une procédure fédérée, le mappage utilisateur et les privilèges de l'ID autorisation ayant créé la procédure fédérée sont utilisés pour accéder aux tables de source de données.

Par exemple, l'utilisateur ZELLER crée une procédure fédérée appelée FP1. Cette dernière référence une procédure Sybase qui accède à une table Sybase. L'ID utilisateur distant du mappage utilisateur ZELLER bénéficie d'un privilège qui lui permet de mettre à jour la table Sybase. L'utilisateur ZELLER octroie le privilège EXECUTE à l'utilisateur BHATIA pour la procédure FP1. L'utilisateur BHATIA doit disposer d'un mappage utilisateur valide par rapport à un ID utilisateur distant qui dispose du privilège EXECUTE pour la procédure Sybase référencée par la procédure FP1. L'ID utilisateur distant vers lequel l'utilisateur BHATIA est mappé n'a pas besoin du privilège SELECT dans la procédure Sybase. Lorsque l'utilisateur BHATIA appelle la procédure FP1, l'utilisateur BHATIA peut mettre à jour la table dans Sybase.

Appels de procédures et niveaux d'accès

Les appels de procédures et les niveaux d'accès sont confrontés aux problèmes suivants :

- Par défaut, la clause CALL RESOLUTION IMMEDIATE est utilisée avec les procédures fédérées lorsque vous émettez la commande PRECOMPILE. La clause CALL RESOLUTION DEFERRED n'est pas prise en charge par les procédures fédérées.
- La sortie n'est pas visible lorsqu'une procédure fédérée appelle une procédure de source de données qui s'imprime dans une mémoire tampon ou une sortie standard.
- Lorsque vous appelez une procédure fédérée à partir d'une fonction externe définie par un utilisateur, le niveau d'accès de cette procédure fédérée ne doit pas être défini sur READS SQL DATA ou sur MODIFIES SQL DATA. L'accès fédéré est bloqué dans les fonctions externes.
- Les restrictions qui s'appliquent à l'appel des procédures locales dans le mode passe-système s'appliquent également aux procédures fédérées.
- Chaque argument de l'instruction CALL doit être compatible avec le paramètre correspondant dans la procédure. Les procédures fédérées sont soumises aux mêmes règles d'affectation des paramètres que les procédures locales.

Transactions

Supposons que vous rencontrez les problèmes suivants lors de l'utilisation de procédures fédérées avec des transactions :

- La procédure de source de données que la procédure fédérée référence ne doit pas émettre d'instruction COMMIT ou ROLLBACK. La fédération ne doit pas appliquer cette restriction et des incohérences de données peuvent se produire si la procédure de source de données émet une instruction COMMIT ou ROLLBACK.
- Les procédures fédérées dotées du niveau d'accès MODIFIES SQL DATA ne peuvent pas être invoquées dans des déclencheurs, des instructions de composants dynamiques, une fonction scalaire SQL, des tables, des fonctions de

ligne et des méthodes. Après avoir émis une instruction SAVEPOINT, vous ne pouvez pas appeler une procédure fédérée dotées du niveau d'accès MODIFIES SQL DATA.

- Les procédures fédérées ne prennent pas en charge les curseurs WITH HOLD, les curseurs flottants ou les curseurs pouvant être mis à jour. Si la procédure de source de données utilise ces types de curseurs, aucun avertissement ou message d'erreur ne s'affiche. Les applications accédant aux ensembles de résultats à l'aide d'une procédure de source de données qui utilise ces types de curseurs peuvent avoir un comportement différent. Les curseurs sont généralement renvoyés sans la fonction de conservation et les curseurs de défilement avant en lecture seule.

Vues de catalogue

Les vues catalogue système suivantes stockent des informations relatives aux procédures fédérées :

- SYSCAT.ROUTINES
- SYSCAT.ROUTINESFEDERATED
- SYSCAT.ROUTINEOPTIONS
- SYSCAT.ROUTINEPARMS
- SYSCAT.ROUTINEPARMOPTIONS

Sources de données et procédures fédérées

Avant de créer des procédures fédérées, contrôlez la manière dont la fédération prend en charge les procédures de source de données.

Procédures fédérées pour les sources de données DB2

Avant de créer une procédure fédérée, contrôlez la méthode de spécification des options dans l'instruction CREATE PROCEDURE.

Lors de la création de procédures fédérées, n'oubliez pas les restrictions suivantes :

- Les procédures DB2 prennent en charge les paramètres IN, OUT et INOUT.
- Vous devez spécifier les mêmes options que celles de la procédure DB2 pour un accès SQL, les clauses d'action externes et déterministes de l'instruction CREATE PROCEDURE.
- Si deux procédures DB2 éloignées ont un nom identique, utilisez l'option NUMBER OF PARAMETERS pour identifier la procédure à utiliser.
- La base de données fédérée n'émet pas l'instruction COMMIT pour les procédures fédérées qui sont créées pour les procédures DB2 Universal Database for z/OS et qui contiennent une clause COMMIT ON RETURN YES.

Exemple

L'exemple suivant indique comment utiliser l'instruction CREATE PROCEDURE qui permet de créer une procédure fédérée pour une procédure de source de données sous DB2.

```
CREATE PROCEDURE PROC1 SOURCE KELLER.PROC1_DB2
    NUMBER OF PARAMETERS 3 FOR SERVER DB2_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Obligatoire. Indique le nom de la procédure fédérée.

SOURCE KELLER.PROC1_DB2

Obligatoire. Indique le schéma et le nom de la procédure DB2. Pour les procédures DB2, spécifiez un nom bipartite dans l'instruction CREATE PROCEDURE. Le format de ce nom est *nom_schéma_source.nom_procédure_source*.

NUMBER OF PARAMETERS 3

Indique le nombre total de paramètres IN, OUT et INOUT que la procédure DB2 utilise. Utilisez ce paramètre lorsque plusieurs procédures affichent le même nom de schéma et de procédure. Par exemple, si votre schéma s'intitule KELLER et que vous disposez d'une procédure PROC1 dotée de trois paramètres et une autre procédure PROC1 dotée d'un paramètre, le nom de ces deux procédures est KELLER.PROC1. La valeur de NUMBER OF PARAMETERS dans la procédure de source de données indique la procédure à laquelle vous vous référez dans l'instruction CREATE PROCEDURE.

FOR SERVER DB2_SERVER

Obligatoire. Indique une définition de serveur dans lequel la procédure fédérée est créée.

SPECIFIC MYPROC1

Indique un nom unique pour la procédure fédérée que vous créez. Ce paramètre est utilisé uniquement pour les procédures fédérées et n'est pas associé aux procédures de source de données. Si vous ne spécifiez pas de nom unique, un nom est alors généré par le gestionnaire de base de données fédérée.

WITH RETURN TO CLIENT ALLITH RETURN TO CLIENT ALL

Indique que l'ensemble de résultats est renvoyé à l'application client. La fédération renvoie un ensemble de résultats au maximum. Si ce paramètre n'est pas spécifié, le paramètre par défaut est WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indique le niveau d'accès aux données des instructions SQL qui sont incluses dans la procédure fédérée. Si la clause spécifiée ne correspond pas à la procédure DB2, un message d'erreur est renvoyé. Si cette clause n'est pas spécifiée, la clause de la procédure DB2 est alors utilisée.

DETERMINISTIC

Indique si la procédure fédérée renvoie toujours les mêmes résultats pour un ensemble de valeurs d'arguments données. Ce paramètre peut améliorer les performances de l'interaction entre le serveur fédéré et la source de données. Si la clause spécifiée ne correspond pas à la procédure DB2, un message d'erreur est renvoyé. Si cette clause n'est pas spécifiée, la clause de la procédure DB2 est alors utilisée.

EXTERNAL ACTION

Indique si la procédure fédérée utilise une action qui modifie l'état d'un objet non géré par le gestionnaire de base de données. Si la clause spécifiée ne correspond pas à la procédure DB2, un message d'erreur est renvoyé. Si cette clause n'est pas spécifiée, la clause de la procédure DB2 est alors utilisée.

Procédures fédérées et Microsoft SQL Server

Avant de créer une procédure fédérée, contrôlez les paramètres pris en charge, la manière dont les ensembles de résultats sont renvoyés et les restrictions existantes.

Paramètres

Les procédures Microsoft SQL Server prennent en charge l'utilisation des paramètres INPUT et OUTPUT facultatifs. L'encapsuleur SQL Server mappe chaque paramètre INPUT vers un paramètre IN fédéré et chaque paramètre OUTPUT vers un paramètre INOUT fédéré. Si vous utilisez des paramètres facultatifs dans une procédure SQL Server, vous devez les compter lorsque vous spécifiez la clause NUMBER OF PARAMETERS de l'instruction CREATE PROCEDURE.

Ensembles de résultats

L'encapsuleur SQL Server peut renvoyer un ensemble de résultats et un paramètre OUTPUT. Lorsqu'une procédure SQL Server renvoie un paramètre OUTPUT et un ensemble de résultats, seul le paramètre est renvoyé. L'ensemble de résultats est supprimé et le message d'erreur SQL0464W s'affiche. La valeur DB2_RETURN_STATUS est récupérée pour les procédures qui renvoient des ensembles de résultats, mais elle renvoie toujours la valeur zéro (0), peu importe la valeur de retour réelle de la procédure.

Restrictions

L'encapsuleur SQL Server ne peut pas appeler de procédure dans les situations suivantes :

- Lorsqu'une autre procédure est déjà appelée
- Lorsqu'une autre instruction est exécutée lors d'une connexion simple

Pour pallier à ces restrictions, vous pouvez définir la procédure fédérée sur un autre serveur. Dans cet exemple, les instructions suivantes ont réussi si le pseudonyme *pseudo_sql* et la procédure *proc_sql* sont définis sur plusieurs serveurs. Si le pseudonyme et la procédure sont définis sur le même serveur, les instructions échouent.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM pseudo_sql OPEN clientcur; CALL sql_proc();
```

Exemple

L'exemple suivant montre comment utiliser l'instruction CREATE PROCEDURE qui permet de créer une procédure fédérée pour une procédure de source de données sous Microsoft SQL Server. Notez que Microsoft SQL Server ne prend pas en charge la clause UNIQUE ID, ainsi que la valeur nom_module_source de la clause SOURCE.

```
CREATE PROCEDURE PROC1 SOURCE BHATIA.PROC1_MSSQL  
    NUMBER OF PARAMETERS 5 FOR SERVER MSSQL_SERVER  
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL  
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Obligatoire. Indique le nom de la procédure fédérée.

SOURCE BHATIA.PROC1_MSSQL

Obligatoire. Indique le nom du schéma et de la procédure sur la source de données.

FOR SERVER MSSQL_SERVER

Obligatoire. Indique une définition de serveur dans lequel la procédure fédérée est créée.

NUMBER OF PARAMETERS 5

Indique le nombre total de paramètres IN et OUTPUT que la procédure de source de données utilise.

SOURCE BHATIA.PROC1_SQL

Indique le schéma et le nom de la procédure de source de données. Le format de ce nom bipartite est *nom_schéma_source.nom_procédure_source*.

SPECIFIC MYPROC1

Indique un nom unique pour la procédure fédérée que vous créez. Ce paramètre est utilisé uniquement pour les procédures fédérées et n'est pas associé aux procédures de source de données. Si vous ne spécifiez pas de nom unique, un nom est alors généré par le gestionnaire de base de données fédérée.

WITH RETURN TO CLIENT ALLITH RETURN TO CLIENT ALL

Indique que l'ensemble de résultats est renvoyé à l'application client. La fédération renvoie un ensemble de résultats au maximum. Si ce paramètre n'est pas spécifié, le paramètre par défaut est WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indique le niveau d'accès aux données des instructions SQL qui sont incluses dans la procédure fédérée. Si la clause spécifiée ne correspond pas à la procédure de source de données, un message d'erreur est renvoyé. Si cette clause n'est pas spécifiée, la clause de la procédure de source de données est alors utilisée.

DETERMINISTIC

Indique si la procédure fédérée renvoie toujours les mêmes résultats pour un ensemble de valeurs d'arguments données. Ce paramètre peut améliorer les performances de l'interaction entre le serveur fédéré et la source de données.

EXTERNAL ACTION

Indique si la procédure fédérée utilise une action qui modifie l'état d'un objet non géré par le gestionnaire de base de données.

Procédures fédérées et Oracle

Vous pouvez créer une procédure fédérée et un mappage de fonction pour la même fonction Oracle.

Utilisez un mappage de fonction lorsque vous devez utiliser la fonction Oracle dans SQL en tant que fonction scalaire. Utilisez une procédure fédérée lorsque vous devez utiliser la fonction Oracle dans les instructions CALL.

Oracle ne renvoie qu'une seule valeur pour les fonctions. La valeur de retour pour la procédure fédérée s'affiche au début de la liste de paramètres en tant que paramètre OUT supplémentaire. Le nom du paramètre est toujours DEFAULT. Lorsque vous spécifiez la clause NUMBER OF PARAMETERS dans l'instruction CREATE PROCEDURE (dérivée), ne comptez pas les valeurs de retour.

Pour certains types de données Oracle, les informations relatives à la précision, la longueur et l'échelle ne sont pas stockées dans le catalogue Oracle lorsque les paramètres d'une procédure sont déclarés. Lors de la création d'une procédure fédérée, des informations sur la procédure Oracle sont regroupées dans le catalogue Oracle. Etant donné que les informations relatives à la précision, la

longueur et l'échelle ne sont pas stockées dans le catalogue Oracle, le comportement des procédures fédérées est le suivant :

- Utilisez la longueur maximale pour les types de données de paramètre.
- Mappez les types de données NUMBER d'Oracle vers le type de données DOUBLE fédérées. Vous pouvez modifier ce mappage en remplaçant le mappage des types de données en aval par défaut pour le type de données NUMBER d'Oracle.

Conseil : Le remplacement des mappages de types de données en aval par défaut affecte d'autres opérations DDL fédérées, telles que CREATE NICKNAME. Par conséquent, avant de créer la procédure, modifiez le mappage de type de données. Créez une procédure pour la procédure Oracle avec le nouveau mappage de type de données, puis supprimez-le. Les pseudonymes et procédures suivants que vous créez vont utiliser le mappage de type de données par défaut.

Exemple

L'exemple suivant montre comment utiliser l'instruction CREATE PROCEDURE qui permet de créer une procédure fédérée pour une procédure de source de données sous Oracle.

```
CREATE PROCEDURE PROC2 SOURCE ZELLER_SCHEMA.ORACLE_PKG9.PROC2
    NUMBER OF PARAMETERS 5 UNIQUE_ID '2' FOR SERVER ORA_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION;
```

PROC2

Obligatoire. Indique le nom de la procédure fédérée.

SOURCE ZELLER_SCHEMA.ORACLE_PKG9.PROC2

Obligatoire. Indique le schéma, le module et le nom de la procédure ou de la fonction Oracle. Si la procédure ou la fonction Oracle se trouve dans un module, vous devez spécifier un nom tripartite dans l'instruction CREATE PROCEDURE. Le format de ce nom est *nom_schéma_source.nom_module_source.nom_procédure_source*. Si la procédure ou la fonction Oracle ne se trouve pas dans un module, vous devez spécifier un nom bipartite dans l'instruction CREATE PROCEDURE. Le format de ce nom est *nom_schéma_source.nom_procédure_source*.

NUMBER OF PARAMETERS 5

Indique le nombre total de paramètres IN, OUT et INOUT utilisés par la procédure Oracle. Utilisez ce paramètre lorsque plusieurs procédures affichent le même nom de schéma et de procédure. Par exemple, si votre schéma s'intitule ZELLER et que vous disposez d'une procédure PROC1 dotée de deux paramètres et d'une autre procédure PROC1 dotée de trois paramètres, le nom de ces deux procédures est ZELLER.PROC1. La valeur de NUMBER OF PARAMETERS dans la procédure de source de données indique la procédure à laquelle vous vous référez dans l'instruction CREATE PROCEDURE. Les paramètres REFCURSOR d'Oracle doivent être inclus dans le décompte NUMBER OF PARAMETERS.

UNIQUE_ID '2'

Indique l'identificateur unique de la procédure Oracle. Utilisez le paramètre UNIQUE_ID uniquement lorsque le nom de schéma, le nom de procédure et le nombre de paramètres ne servent pas uniquement à identifier une procédure Oracle. Le paramètre UNIQUE ID correspond à la valeur affichée dans la colonne ALL_ARGUMENTS.OVERLOAD du catalogue système d'Oracle. Si ce paramètre n'est pas spécifié, le serveur

fédéré détecte les procédures multiréférencées et renvoie une erreur. Utilisez cette option uniquement avec les procédures Oracle.

FOR SERVER ORA_SERVER

Obligatoire. Indique une définition de serveur dans lequel la procédure fédérée est créée.

SPECIFIC MYPROC1

Indique un nom unique pour la procédure fédérée que vous créez. Ce paramètre est utilisé uniquement pour les procédures fédérées et n'est pas associé aux procédures de source de données. Si vous ne spécifiez pas de nom unique, un nom est alors généré par le gestionnaire de base de données fédérée. Ce paramètre est facultatif.

WITH RETURN TO CLIENT ALLITH RETURN TO CLIENT ALL

Indique que l'ensemble de résultats est renvoyé à l'application client. La fédération renvoie un ensemble de résultats au maximum. Si ce paramètre n'est pas spécifié, le paramètre par défaut est WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indique le niveau d'accès aux données des instructions SQL qui sont incluses dans la procédure fédérée. Si la clause spécifiée ne correspond pas à la procédure Oracle, un message d'erreur est renvoyé. Si elle n'est pas spécifiée, la clause de la procédure de source de données est alors utilisée.

DETERMINISTIC

Indique si la procédure fédérée renvoie toujours les mêmes résultats pour un ensemble de valeurs d'arguments données. Ce paramètre peut améliorer les performances de l'interaction entre le serveur fédéré et la source de données.

NO EXTERNAL ACTION

Indique si la procédure fédérée utilise une action qui modifie l'état d'un objet non géré par le gestionnaire de base de données.

Procédures multiréférencées dans les systèmes fédérés

Les procédures *multiréférencées* sont des procédures qui disposent de noms et de schémas identiques. Elles peuvent contenir un nombre différents de paramètres ou plusieurs signatures de paramètres. L'objectif d'une procédure multiréférencée est de créer des versions semblables d'une procédure.

Le serveur fédéré autorise des procédures multiréférencées uniquement si chaque procédure dispose d'un nombre différent de paramètres.

Oracle autorise des procédures multiréférencées si chaque procédure dispose d'un nombre différent de paramètres ou si les types de paramètres sont différents. Vous pouvez créer des procédures fédérées pour les procédures multiréférencées Oracle.

Pour différencier des procédures qui utilisent un nom identique, un nom de schéma et plusieurs paramètres, vous devez spécifier UNIQUE ID lors de la création d'une procédure fédérée.

Vous pouvez déterminer UNIQUE ID de deux façons différentes, en utilisant la fonction de reconnaissance dans le Centre de contrôle et en interrogeant le catalogue Oracle.

Utilisation de la fonction de reconnaissance du Centre de contrôle

Lors de la création d'une procédure fédérée dans le Centre de contrôle, la fonction de reconnaissance détermine si la procédure Oracle est multiréférencée. La valeur multiréférencée est affichée dans la zone UNIQUE ID de chaque procédure Oracle.

Interrogation du catalogue Oracle

Pour déterminer le paramètre UNIQUE ID, vous pouvez interroger la colonne OVERLOAD dans le catalogue SYS.ALL_ARGUMENTS d'Oracle. La valeur UNIQUE ID est un littéral de type caractère qui contient un numéro, 1 par exemple. Vous pouvez utiliser le mode passe-système sur le serveur fédéré ou interroger directement le client Oracle.

Par exemple, pour interroger le catalogue Oracle, afficher la signature et multiréférencer la colonne d'une procédure commençant par HJZ, utilisez l'instruction SELECT suivante :

```
SELECT owner, package_name, object_name, overload, position, argument_name, in_out,
data_type
FROM all_arguments aa
WHERE object_name like 'HJZ%'
ORDER BY owner, package_name, object_name, overload, position;
```

La sortie suivante issue de la requête ci-dessus indique que le module HJZ_PACK1 contient trois procédures utilisant le nom HJZTEST1. Déterminez le nombre de procédures en observant les colonnes OBJECT_NAME AND OVERLOAD. La première procédure contient un paramètre IN avec un type de données Number. La deuxième procédure contient un paramètre IN avec un type de données Character. La troisième procédure contient un paramètre OUT avec un type de données Character et un paramètre IN avec un type de données Number. La sortie indique que deux procédures utilisent le nom HJZTEST3 dans le module HJZ_PACK1. Il existe également une procédure appelée HJZTEST1 qui ne fait pas partie du module. Cette dernière procédure contient un paramètre IN qui utilise un type de données Number.

OWNER	PACKAGE_NAME	OBJECT_NAME	OVERLOAD	POSITION	ARGUMENT_NAME	IN_OUT	DATA_TYPE
J15USER1	HJZ_PACK1	HJZTEST1	1	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST1	2	1	A	IN	CHAR
J15USER1	HJZ_PACK1	HJZTEST1	3	0	-	OUT	CHAR
J15USER1	HJZ_PACK1	HJZTEST1	3	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	1	1	A	IN	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	1	2	B	OUT	NUMBER
J15USER1	HJZ_PACK1	HJZTEST3	2	1	A	IN	CHAR
J15USER1	HJZ_PACK1	HJZTEST3	2	2	B	OUT	CHAR
J15USER1	-	HJZTEST1	-	1	A	IN	NUMBER

9 enregistrement(s) sélectionnés.

Pour créer une procédure fédérée pour la deuxième procédure multiréférencée dotée d'un paramètre IN avec un type de données CHAR, émettez l'instruction CREATE PROCEDURE suivante :

```
CREATE PROCEDURE HJZTEST1 SOURCE J15USER1.HJZ_PACK1.HJZTEST1
NUMBER OF PARAMETERS 1 UNIQUE ID '2'
FOR SERVER ORA_SERVER WITH RETURN TO CLIENT ALL;
```

Important : Dans l'exemple ci-dessus, la clause NUMBER OF PARAMETERS n'identifie pas uniquement la procédure. Deux procédures HJZTEST1 sont disponibles dans la table et chacune dispose d'un paramètre. Vous devez spécifier la clause UNIQUE ID pour indiquer la procédure multiréférencée à utiliser. Utilisez

la valeur issue de la colonne OVERLOAD comme valeur pour la clause UNIQUE_ID. Lorsque la clause UNIQUE ID est spécifiée, la clause NUMBER OF PARAMETERS est facultative. Utilisez la clause NUMBER OF PARAMETERS pour confirmer que la procédure de source de données dispose du nombre de paramètres prévus.

Procédures fédérées et Sybase

Avant de créer une procédure fédérée, contrôlez les paramètres pris en charge, la manière dont les ensembles de résultats sont renvoyés et les restrictions existantes.

Paramètres

Les procédures Sybase utilisent les paramètres INPUT et OUTPUT. L'encapsuleur Sybase mappe un paramètre INPUT Sybase vers un paramètre IN fédéré et un paramètre OUTPUT Sybase vers un paramètre INOUT fédéré. Même si vous pouvez utiliser des paramètres facultatifs dans une procédure Sybase, vous ne pouvez pas les utiliser dans une procédure fédérée. Par conséquent, lors de l'émission d'une instruction CALL, vous devez spécifier tous les paramètres.

Pour Sybase Version 12.0, tous les paramètres sont des paramètres d'entrée. Les procédures fédérées ne peuvent pas renvoyer une valeur de paramètre de sortie. Il s'agit d'une restriction du catalogue Sybase, celle-ci ne s'applique pas aux versions ultérieures de Sybase.

Ensembles de résultats

Pour les procédures Sybase qui renvoient un paramètre de sortie et un ensemble de résultats, ce dernier est supprimé. Si la procédure Sybase renvoie un ensemble de résultats et une valeur de retour, une valeur de retour égale à 0 est fournie, peu importe la valeur de retour réelle issue de la procédure de source de données. Aucun message d'avertissement ne s'affiche dans ces cas-là.

Toutefois, les ensembles de résultats sont supprimés et le message SQL0464W est renvoyé lorsque les deux conditions suivantes sont vraies :

- La procédure fédérée est définie pour une procédure Sybase qui renvoie des ensembles de résultats.
- La procédure fédérée est invoquée dans un déclencheur ou une fonction définie par un utilisateur.

Restrictions

L'encapsuleur Sybase ne peut pas appeler une procédure dans les situations suivantes :

- Lorsqu'une autre procédure est déjà appelée
- Lorsqu'une autre instruction est exécutée lors d'une connexion simple

Pour pallier à ces restrictions, vous pouvez définir la procédure fédérée sur un autre serveur. Dans l'exemple Sybase suivant, les instructions suivantes ont réussi si le pseudonyme *pseudo_syb* et la procédure *proc_syb* sont définis sur plusieurs serveurs. Si le pseudonyme et la procédure sont définis sur le même serveur, les instructions échouent.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM pseudo_syb OPEN clientcur; CALL syb_proc();
```

Exemple

L'exemple suivant montre comment utiliser l'instruction CREATE PROCEDURE qui permet de créer une procédure fédérée pour une procédure de source de données sous Sybase. Notez que Sybase ne prend pas en charge la clause UNIQUE ID de l'instruction CREATE PROCEDURE.

```
CREATE PROCEDURE PROC1 SOURCE BHATIA.PROC1_SYBASE
    NUMBER OF PARAMETERS 3 FOR SERVER SYBASE_SERVER
    SPECIFIC MYPROC1 WITH RETURN TO CLIENT ALL
    MODIFIES SQL DATA DETERMINISTIC EXTERNAL ACTION;
```

PROC1

Obligatoire. Indique le nom de la procédure fédérée.

SOURCE BHATIA.PROC1_SYBASE

Obligatoire. Indique le schéma et le nom de la procédure Sybase. Pour les procédures Sybase, spécifiez un nom bipartite dans l'instruction CREATE PROCEDURE. Le format de ce nom est *nom_schéma_source.nom_procédure_source*.

NUMBER OF PARAMETERS 3

Indique le nombre total de paramètres IN, OUT et INOUT que la procédure Sybase utilise. Utilisez ce paramètre lorsque plusieurs procédures affichent le même nom de schéma et de procédure. Par exemple, si votre schéma s'intitule BHATIA et que vous disposez d'une procédure PROC1 dotée de trois paramètres et d'une autre procédure PROC1 dotée d'un paramètre, le nom de ces deux procédures est BHATIA.PROC1. La valeur de NUMBER OF PARAMETERS dans la procédure de source de données indique la procédure à laquelle vous vous référez dans l'instruction CREATE PROCEDURE.

FOR SERVER SYBASE_SERVER

Obligatoire. Indique une définition de serveur dans lequel la procédure fédérée est créée.

SPECIFIC MYPROC1

Indique un nom unique pour la procédure fédérée que vous créez. Ce paramètre est utilisé uniquement pour les procédures fédérées et n'est pas associé aux procédures de source de données. Si vous ne spécifiez pas de nom unique, un nom est alors généré par le gestionnaire de base de données fédérée.

WITH RETURN TO CLIENT ALLITH RETURN TO CLIENT ALL

Indique que l'ensemble de résultats est renvoyé à l'application client. La fédération renvoie un ensemble de résultats au maximum. Si ce paramètre n'est pas spécifié, le paramètre par défaut est WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indique le niveau d'accès aux données des instructions SQL qui sont incluses dans la procédure fédérée. Si la clause spécifiée ne correspond pas à la procédure Sybase, un message d'erreur est renvoyé. Si elle n'est pas spécifiée, la clause de de la procédure Sybase est utilisée.

DETERMINISTIC

Indique si la procédure fédérée renvoie toujours les mêmes résultats pour un ensemble de valeurs d'arguments données. Ce paramètre peut améliorer les performances de l'interaction entre le serveur fédéré et la source de données.

EXTERNAL ACTION

Indique si la procédure fédérée utilise une action qui modifie l'état d'un objet non géré par le gestionnaire de base de données.

Création de procédures fédérées

Vous devez créer une procédure fédérée pour chaque procédure de source de données à appeler depuis le serveur fédéré.

Avant de commencer

- La procédure de source de données doit exister au préalable.
- L'encapsuleur de la source de données a été enregistré et la définition de serveur a été créée.
- Tous les mappages utilisateur requis ont été créés.
- L'ID autorisation de l'instruction contient au moins l'un des privilèges suivants :
 - Privilège IMPLICIT_SCHEMA sur la base données, si le nom de schéma de la procédure ne fait pas référence à un schéma existant
 - Privilège CREATEIN sur le schéma, si le nom de schéma de la procédure fait référence à un schéma existant
 - droit SYSADM ou DBADM
- Les privilèges de l'ID autorisation dont dispose la source de données doivent également inclure le privilège permettant de sélectionner la description de la procédure à partir des tables éloignées du catalogue.

A propos de cette tâche

Pour créer une procédure fédérée, utilisez l'instructions SQL CREATE PROCEDURE (dérivée) ou le Centre de contrôle. Lorsque vous utilisez le Centre de contrôle, sélectionnez une procédure de source de données et le Centre de contrôle récupère automatiquement les informations requises pour créer la procédure fédérée.

Pour utiliser le Centre de contrôle, procédez comme suit :

1. Cliquez avec le bouton droit de la souris sur le dossier **Procédures mémorisées fédérées**, puis cliquez sur **Créer**. Le dossier **Procédures mémorisées fédérées** se trouve sous l'encapsuleur et sous la définition de serveur enregistrés pour la source de données.
2. Dans la fenêtre **Créer une procédure mémorisée fédérée**, cliquez sur **Reconnaître**.
3. Dans la fenêtre **Reconnaître**, spécifiez les critères permettant de filtrer les recherches par schéma, nom de module, nom de procédure ou nom de fonction. Utilisez les opérateurs BETWEEN, IN, NOT IN et LIKE afin de spécifier les critères et placez des guillemets simples autour des valeurs spécifiées. Le bouton **Compter** figurant dans la fenêtre **Reconnaître** permet de déterminer le nombre de procédures qui vont être renvoyées. Si la valeur est élevée, spécifiez des critères supplémentaires pour limiter la recherche.
4. Facultatif : Dans la fenêtre **Créer une procédure mémorisée fédérée**, cochez la case en regard de la procédure que vous voulez créer, puis cliquez sur **Propriétés** pour vérifier les informations et veillez à ce que toutes les zones obligatoires soient renseignées avant de créer la procédure fédérée.
5. Dans la fenêtre **Créer une procédure mémorisée fédérée**, cochez la case en regard d'une ou de plusieurs procédures que vous voulez créer, puis cliquez sur **OK**.

Octroi ou révocation d'autorisations d'appel de procédures fédérées

L'administrateur de la base de données fédérée doit octroyer les autorisations requises aux autres utilisateurs afin qu'ils puissent appeler des procédures fédérées.

Avant de commencer

L'utilisateur qui appelle la procédure fédérée doit disposer d'un mappage utilisateur valide entre le serveur fédérée et la source de données. L'ID utilisateur distant doit disposer d'une autorisation sur la source de données équivalente à l'autorisation EXECUTE sur le serveur fédéré. Un utilisateur peut se voir octroyer une autorisation EXECUTE sur la procédure fédérée. Mais si l'autorisation de l'utilisateur sur la source de données n'est pas équivalente à l'autorisation EXECUTE sur le serveur fédéré, les appels effectués vers la procédure de source de données échouent.

L'ID autorisation de l'instruction GRANT doit disposer de l'un des droits suivants :

- L'option WITH GRANT OPTION de l'autorisation EXECUTE sur la procédure fédérée
- droit SYSADM ou DBADM

Procédure

Pour octroyer ou révoquer l'autorisation d'appel de procédures fédérées, procédez comme suit :

Spécifiez les privilèges liés à une autorisation en utilisant le Centre de contrôle ou à partir de la ligne de commande :

Méthode	Description
Centre de contrôle	<ol style="list-style-type: none">1. Cliquez sur le bouton droit de la souris sur le nom de la procédure fédérée, puis cliquez sur Privilèges.2. Sélectionnez l'utilisateur ou le groupe auquel vous souhaitez attribuer des privilèges. Pour ajouter un nouvel utilisateur, cliquez sur Ajouter un utilisateur. Pour ajouter un nouveau groupe, cliquez sur l'onglet Groupe, puis sur Ajouter un groupe.3. Dans la liste déroulante Privilèges : EXECUTE :<ul style="list-style-type: none">• Sélectionnez Oui pour octroyer uniquement le privilège EXECUTE.• Sélectionnez Accorder pour octroyer les privilèges EXECUTE et WITH GRANT OPTION.• Sélectionnez Non pour supprimer le privilège EXECUTE de l'utilisateur ou du groupe.4. Vous pouvez octroyer ou révoquer simultanément des privilèges à différents utilisateurs. Sélectionnez les utilisateurs dans la liste et cliquez sur Tout accorder ou Tout révoquer afin d'accorder ou révoquer le privilège EXECUTE pour les utilisateurs sélectionnés.5. Cliquez sur OK.

Méthode	Description
Ligne de commande	<p>Spécifiez les privilèges dans l'instruction GRANT.</p> <p>Exemple 1 : Pour octroyer le privilège EXECUTE sur toutes les procédures dans le schéma BHATIA, y compris les procédures créées ultérieurement, aux utilisateurs du groupe HR_DEPT, utilisez l'instruction suivante : <pre>GRANT EXECUTE ON PROCEDURE BHATIA.* TO HR_DEPT</pre></p> <p>Exemple 2 : Pour octroyer le privilège EXECUTE sur la procédure PROC1 à l'utilisateur ZELLER et lui permettre d'octroyer le privilège EXECUTE sur cette procédure à d'autres utilisateurs, utilisez l'instruction suivante : <pre>GRANT EXECUTE ON PROCEDURE PROC1 TO ZELLER WITH GRANT OPTION</pre></p> <p>Exemple 3 : Pour octroyer le privilège EXECUTE à l'utilisateur ERFAN sur la procédure PROC2 qui a été créée avec le nom spécifique MY_PROC2, utilisez l'instruction suivante : <pre>GRANT EXECUTE ON SPECIFIC PROCEDURE MY_PROC2 TO ERFAN</pre></p>

Affichage des informations relatives à un paramètre et appel de procédures fédérées

Utilisez le Centre de contrôle ou interrogez la vue catalogue SYSCAT.ROUTINESFEDERATED pour afficher des informations relatives à un paramètre. Utilisez ensuite l'instructions CALL pour appeler une procédure fédérée.

Avant de commencer

L'utilisateur qui appelle une procédure fédérée doit disposer du privilège EXECUTE sur la source de données et d'un mappage utilisateur valide vers l'ID utilisateur d'une source de données éloignée dotée d'un privilège lui permettant d'accéder aux tables de la source de données.

Pour afficher les informations relatives à un paramètre et appeler une procédure fédérée, utilisez l'une des méthodes suivantes :

Méthode	Procédure
Centre de contrôle	<ol style="list-style-type: none"> Développez les dossiers Objets fédérés, Définitions de serveur et Procédures mémorisées fédérées dans l'arborescence d'objets. Sélectionnez le nom de la procédure fédérée pour afficher des informations, comme le type de données et le mode de chaque paramètre. Cliquez sur OutilsEditeur de commande pour ouvrir l'Editeur de commande et émettre une instruction CALL.
Ligne de commande	<ol style="list-style-type: none"> Utilisez l'instruction SELECT pour interroger la vue SYSCAT.ROUTINEPARMS dans le catalogue de la base de données fédérée. Emettez une instruction CALL.

L'exemple suivant utilise une instruction SELECT pour obtenir des informations sur la procédure fédérée. Par exemple, si la procédure fédérée FEDPROC1 se trouve dans le schéma fédéré BOB, émettez l'instruction SELECT suivante :

```
SELECT ordinal, char(paramname,30)
AS name,
       rowtype, char(typename,30) AS type
FROM syscat.routineparms
WHERE routinename='FEDPROC1' AND
       routineschema = 'BOB'
ORDER BY ordinal;
```

Le résultat de la requête affiche les paramètres suivants :

```
ORDINAL NAME ROWTYPE TYPE
-----
1         P1      P      INTEGER
2         P2      O      VARCHAR
```

Le type de ligne P indique un paramètre d'entrée. Le type de ligne O indique un paramètre de sortie. Le type de ligne B indiquant un paramètre INOUT ne figure pas dans cet exemple.

Modification ou suppression de procédures fédérées

Vous pouvez modifier une procédure fédérée existante en changeant le type de données d'un ou de plusieurs paramètres de la procédure fédérée. Vous ne pouvez pas directement modifier la procédure fédérée pour effectuer des modifications supplémentaires.

Avant de commencer

L'ID autorisation de l'instruction DROP PROCEDURE doit disposer de l'un des droits suivants :

- droit SYSADM ou DBADM
- Privilège DROPIN sur le schéma de la procédure fédérée

- Concepteur de la procédure comme celle enregistrée dans la colonne DEFINER de la vue catalogue de la procédure fédérée
- Privilège CONTROL sur la procédure fédérée

A propos de cette tâche

En plus de modifier le type de données d'un paramètre, vous pouvez effectuer d'autres modifications dans une procédure fédérée. Par exemple, vous pouvez modifier une procédure fédérée si la procédure éloignée change. Dans ce cas, vous ne pouvez pas directement modifier la procédure fédérée. Vous devez d'abord supprimer la procédure, puis la recréer à l'aide de nouveaux paramètres. Sinon, vous devez remplacer la procédure existante à l'aide de l'instruction CREATE OR REPLACE PROCEDURE.

Lorsque vous supprimez une procédure fédérée, elle est supprimée du catalogue système sur la base de données fédérée, mais la procédure de source de données n'est pas affectée. N'oubliez pas que lorsque vous supprimez une procédure fédérée, les applications qui sont dépendantes des procédures supprimées peuvent être sensiblement affectées.

Vous pouvez utiliser le Centre de contrôle ou la ligne de commande pour supprimer une procédure fédérée.

Procédure

Pour cela, utilisez l'une des méthodes suivantes :

Méthode	Procédure
Centre de contrôle	<ol style="list-style-type: none"> 1. Développez les dossiers Objets fédérés, Définitions de serveur et Procédures mémorisées fédérées dans l'arborescence d'objets. 2. Cliquez avec le bouton droit de la souris sur la procédure fédérée à supprimer, puis cliquez sur Supprimer.
Ligne de commande	Utilisez l'instruction DROP. Par exemple : DROP PROCEDURE <i>nom_procédure_fédérée</i>

Jointure d'ensembles de résultats pour les procédures fédérées

Vous pouvez joindre des ensembles de résultats renvoyés par une procédure fédérée à l'aide de la commande DB2FEDGENTF.

Avant de commencer

- Pour utiliser le paramètre **-create** de la commande DB2FEDGENTF, vous devez disposer du droit DBADM sur la base de données fédérée ou d'une combinaison des droits ou privilèges suivants sur la base de données fédérée.
 1. Vous devez disposer de l'un des droits suivants :
 - CREATE_EXTERNAL_ROUTINE
 - CREATETAB
 2. Vous devez également disposer de l'un des droits ou privilèges suivants :
 - Rédigez le privilège pour le répertoire *rep_install*/sqlib/function, où *rep_install* est le répertoire dans lequel le serveur fédéré est installé

- Droit IMPLICIT_SCHEMA si le nom du schéma implicite ou explicite de la fonction n'existe pas
- Privilège CREATEIN sur le schéma si le nom de schéma de la fonction existe
- Pour utiliser le paramètre **-drop** de la commande DB2FEDGENTF, vous devez disposer de l'un des droits ou privilèges suivants sur la base de données fédérée :
 - Privilège DROPIN sur le schéma pour l'objet
 - OWNER de l'objet
 - droit DBADM

A propos de cette tâche

Joignez les ensembles de résultats renvoyés par les procédures fédérées pour joindre les données issues des tables locales et éloignées, plus particulièrement des systèmes où l'accès est uniquement autorisé via des procédures fédérées.

Procédure

Pour joindre des ensembles de résultats de procédures fédérées, procédez comme suit :

1. Créez et enregistrez une fonction de table avec le paramètre **-create** de la commande DB2FEDGENTF, par exemple :


```
DB2FEDGENTF -db sample -u user1 -p password1
             -create
             -stpn S1_INVENTORY
             -tfn S1_INVTRY_TF
             -c 'PART_NUM CHAR(10), S1_QTY INT'
```
2. Utilisez une jointure pour combiner les données dans les ensembles de résultats d'une procédure fédérée. Vous pouvez joindre un ensemble de résultats d'une procédure fédérée à l'aide d'une table locale ou joindre des ensembles de résultats issus de deux procédures fédérées.

Exemples

Dans les exemples suivants, les procédures mémorisées ProINVENTORY et ProINVENTORY2 existent et chacune renvoie l'inventaire de deux fournisseurs sous la forme d'ensembles de résultats. La table PARTS existe également et contient l'inventaire du fabricant avec les noms et types de colonnes suivants :

Tableau 6. Colonnes de la table PARTS

Nom de colonne	Type de colonne
PART_NUM	CHAR(10)
PART_DESC	VARCHAR(400)
INVENTORY	INT

Exemple de jointure d'ensembles de résultats à des tables locales

Dans exemple suivant, l'inventaire d'un fournisseur et d'un fabricant est combiné afin de déterminer l'inventaire total disponible :

1. Utilisez l'instruction CREATE PROCEDURE pour créer la procédure fédérée S1_INVENTORY à partir de la procédure mémorisée ProINVENTORY :

```
CREATE PROCEDURE S1_INVENTORY
SOURCE ProINVENTORY
FOR SERVER ORA1;
```

2. Utilisez la commande DB2FEDGENTF pour créer la fonction de table S1_INVTRY_TF qui contient l'ensemble de résultats PART_NUM et S1_QTY :

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S1_INVENTORY
  -tfm S1_INVTRY_TF
  -c 'PART_NUM CHAR(10), S1_QTY INT'
```

3. Utilisez une jointure pour combiner les données de la table PARTS à l'ensemble de résultats de la fonction de table S1_INVTRY_TF :

```
SELECT a.PART_NUM, a.PART_DESC, a.INVENTORY + b.S1_QTY as MAX_QTY
FROM PARTS a, TABLE(S1_INVTRY_TF()) b
WHERE a. PART_NUM = b. PART_NUM
```

Exemple de jointure de deux ensembles de résultats

Dans l'exemple suivant, le fabricant combine l'inventaire de deux fournisseurs afin de déterminer l'inventaire disponible total :

1. Créez la fonction de table S1_INVTRY_TF pour le premier fournisseur :
 - a. Utilisez l'instruction CREATE PROCEDURE pour créer la procédure fédérée S1_INVENTORY à partir de la procédure mémorisée ProINVENTORY :

```
CREATE PROCEDURE S1_INVENTORY
SOURCE ProINVENTORY
FOR SERVER ORA1;
```

- b. Utilisez la commande DB2FEDGENTF pour créer la fonction de table S1_INVTRY_TF qui contient l'ensemble de résultats PART_NUM et S1_QTY :

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S1_INVENTORY
  -tfm S1_INVTRY_TF
  -c 'PART_NUM CHAR(10), S1_QTY INT'
```

2. Créez la fonction de table S2_INVTRY_TF pour le deuxième fournisseur :

- a. Utilisez l'instruction CREATE PROCEDURE pour créer la procédure fédérée S2_INVENTORY à partir de la procédure mémorisée ProINVENTORY2 :

```
CREATE PROCEDURE S2_INVENTORY
SOURCE ProINVENTORY2
FOR SERVER SYBA1;
```

- b. Utilisez la commande DB2FEDGENTF pour créer la fonction de table S2_INVTRY_TF qui contient l'ensemble de résultats PART_NUM et S2_QTY :

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S2_INVENTORY
  -tfm S2_INVTRY_TF
  -c 'PART_NUM CHAR(10), S2_QTY INT'
```

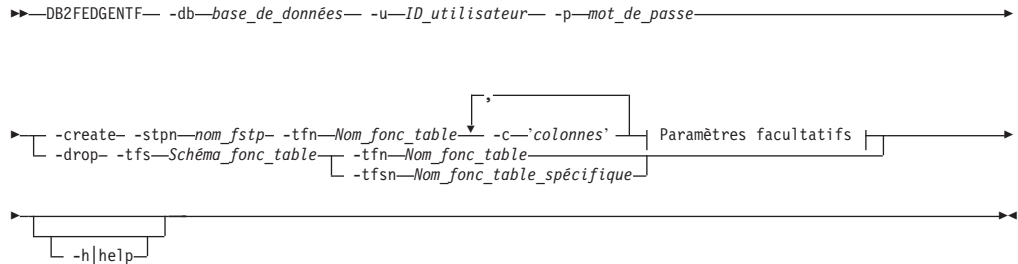
3. Utilisez une jointure pour combiner les ensembles de résultats des fonctions de tables S1_INVTRY_TF et S2_INVTRY_TF :

```
SELECT a.PART_NUM, a.PART_DESC, b.S1_QTY + c.S2_QTY as MAX_SUPP_QTY
FROM PARTS a, TABLE(S1_INVTRY_TF()) b, TABLE(S2_INVTRY_TF()) c
WHERE a. PART_NUM = b. PART_NUM
AND a. PART_NUM = c. PART_NUM
```

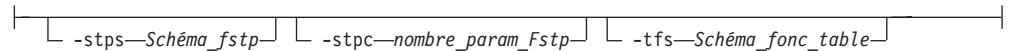
Commande DB2FEDGENTF

Crée et enregistre les fonctions de table qui renvoient des ensembles de résultats depuis des procédures fédérées.

Syntaxe



Paramètres facultatifs :



Paramètres

-db *base_de_données*

Indique le nom de la base de données à laquelle vous voulez vous connecter.

-u *ID_utilisateur*

Indique l'ID utilisateur de la base de données fédérée.

-p *mot_de_passe*

Indique le mot de passe de l'ID utilisateur.

-create

Crée et enregistre une fonction de table dans le schéma actuel si le paramètre **-tfs** n'est pas spécifié. La fonction de table renvoie les colonnes spécifiées de l'ensemble de résultats à partir de la procédure fédérée.

-stpn *Nom_fstp*

Indique le nom de la procédure fédérée.

-tfn *Nom_fonc_table*

Indique le nom de la fonction de table. S'il n'est pas spécifié, le nom de la procédure fédérée est alors utilisé.

-c '*colonnes*'

Indique une liste délimitée par des virgules contenant le nom de la colonne et les paires de types de colonnes dans la signature de l'ensemble de résultats renvoyée par la procédure fédérée.

Exemple

Les noms de colonnes sont PID, PRICE et QTY et les types de colonnes sont CHAR(10), DOUBLE et INT :

```
'PID CHAR(10), PRICE DOUBLE, QTY INT'
```

-stps *Schéma_fstp*

Indique le schéma de la procédure fédérée. Ce paramètre est facultatif. Si un nom n'est pas spécifié, le schéma SQL par défaut défini dans le registre spécial CURRENT SCHEMA est alors utilisé.

-stpc *nombre_param_Fstp*

Indique le nombre d'entrées pour la procédure fédérée. Ce paramètre est facultatif. S'il n'est pas spécifié, la procédure fédérée est déterminée par le nom spécifié de la procédure fédérée. Si les procédures fédérées sont multiréférencés, une erreur s'affiche.

-tfs *Schéma_fonc_table*

Indique le schéma de la fonction table. Ce paramètre est facultatif. S'il n'est pas spécifié, le schéma SQL par défaut défini dans le registre spécial CURRENT SCHEMA est alors utilisé.

- drop

Supprime la fonction de table spécifiée. La description du catalogue est également supprimée et tous les modules qui référencent la fonction de table spécifiée deviennent non valides.

-tfs *Schéma_fonc_table*

Indique le schéma de la fonction de table à supprimer. S'il n'est pas spécifié, le schéma SQL par défaut défini dans le registre spécial CURRENT SCHEMA est alors utilisé.

-tfn *Nom_fonc_table*

Indique le nom de la fonction de table à supprimer.

-tfsn *Nom_fonc_table_spécifique*

Pour les fonctions multiréférencées, indique le nom spécifique de la fonction de table à supprimer. Ce paramètre est mutuellement exclusif avec le paramètre **-tfn** *Nom_fonc_table_*. Vous n'avez pas besoin de spécifier cette option si la fonction de table est uniquement identifiée par son nom et son schéma. Ce paramètre est facultatif.

-h | help

Fournit des information d'utilisation de la commande DB2FEDGENTF.

Exemples

Dans l'exemple suivant, la commande DB2FEDGENTF s'exécute sur la procédure fédérée EAST_INVTRY pour créer la fonction de table S1_INVTRY_TF qui renvoie les colonnes PRODID, PRICE et QTY :

```
DB2FEDGENTF -db sample -u user1 -p password1
- create
- stpn EAST_INVTRY
- tfn E_INVTRY_TF
- c 'PRODID INT, PRICE DOUBLE, QTY INT'
```

Identification et résolution d'incidents dans une procédure fédérée

Si des incidents se produisent dans les procédures fédérées, plusieurs solutions permettent de les résoudre.

Les requêtes et outils de diagnostic suivants permettent d'afficher des informations sur les procédures fédérées. Elles vont vous aider à résoudre les incidents relatifs aux procédures fédérées.

Vérification des informations relatives à une procédure de source de données

Si l'erreur SQL1253N est renvoyée lorsque vous émettez une instruction CREATE PROCEDURE, vous pouvez émettre les requêtes suivantes sur les tables du

catalogue de la source de données pour vérifier les informations relatives à la procédure de source de données. L'erreur SQL1253N indique que la procédure source spécifiée dans l'instruction CREATE PROCEDURE (dérivée) est introuvable dans la source de données. Vous pouvez directement interroger le serveur Oracle ou utiliser une session passe-système.

Pour les procédures DB2 for Linux, UNIX et Windows :

```
SELECT parm_count, result_sets,
       sql_data_access, deterministic, external_action
FROM syscat.routines
WHERE routineschema = ''
AND routinename = ''
AND routinetype = 'P'
AND parm_count = '' <-- optional
```

Pour les procédures DB2 for iSeries :

```
SELECT in_parms+out_parms+inout_parms,
       number_of_results, sql_data_access, deterministic,
       external_action
FROM qsys2.sysroutines
WHERE routine_schema = ''
and routine_name = ''
and routine_type = 'PROCEDURE'
and in_parms+out_parms+inout_parms = ''; <-- optional
```

Pour les procédures DB2 for z/OS :

```
SELECT parm_count, result_sets,
       sql_data_access, deterministic, external_action
FROM sysibm.sysroutines
WHERE schema = ''
AND name = ''
AND routinetype = 'P'
```

Pour Microsoft SQL Server

```
SELECT id
FROM dbo.sysobjects
WHERE id = object_id AND
      (TYPE = 'P' or TYPE = 'X')
```

Pour les procédures Oracle qui se trouvent dans un module :

```
SELECT owner, package_name, object_name, overload, parm_count
FROM (
  SELECT owner, package_name, object_name, overload,
         SUM(case
              WHEN data_type IS NULL
              THEN 0
              ELSE 1
              END)
         AS parm_count
  FROM sys.all_arguments
  WHERE data_level = 0
  GROUP BY owner, package_name, object_name, overload
) aa
WHERE object_name = '' AND
      package_name = '' AND
      owner = '' AND
      overload = '' AND <-- optional
      parm_count =; <-- optional
```

Pour les procédures Oracle qui ne se trouvent pas dans un module :

```
SELECT object_name, object_type, status
FROM sys.all_objects
WHERE owner = '' AND
      object_name = '' AND
      object_type IN ('PROCEDURE', 'FUNCTION')
```

Pour les procédures Sybase :

```
SELECT id
FROM dbo.sysobjects
WHERE id = object_id('.') AND
      (TYPE = 'P' OR TYPE = 'XP')
```

Outils de diagnostic

Utilisez l'utilitaire Explain, la commande DESCRIBE ou db2audit pour diagnostiquer des incidents dans les procédures fédérées.

Par exemple, la procédure FED_PROC1 dispose de trois paramètres OUTPUT. Pour utiliser la commande DESCRIBE sur la procédure FED_PROC1, émettez la commande suivante :

```
DESCRIBE CALL FED_PROC1(?,?,?);
```

Moniteur système

Les éléments du moniteur système de la base de données fédérée contiennent des informations sur les procédures fédérées. Ces éléments sont les suivants :

- L'élément du moniteur Délai de la procédure mémorisée, `stored_proc_time`, indique le délai nécessaire à la source de données pour répondre aux instructions de la procédure fédérée.
- L'élément du moniteur Lignes renvoyées par les procédures mémorisées, `sp_rows_selected`, indique le nombre de lignes envoyées depuis la source de données jusqu'au serveur fédéré. Cet élément permet de calculer le nombre moyen de lignes envoyées au serveur fédéré depuis la source de données pour chaque procédure fédérée ou de calculer le délai moyen requis pour renvoyer une ligne au serveur fédéré depuis la source de données.
- L'élément du moniteur Procédures mémorisées, `stored_procs`, indique le nombre total de procédures que le serveur fédéré a appelé depuis cette source de données.

Erreur SQL SQL30090N avec code retour 21

Il existe plusieurs situations dans lesquelles une erreur SQL30090N avec le code retour 21 est renvoyée. L'une des situations les plus fréquentes se produit lorsqu'une procédure fédérée est créée à l'aide d'un encapsuleur isolé. Des procédures fédérées peuvent être créées uniquement sur des encapsuleurs fiables.

Ensemble de résultats non renvoyés

Un ensemble de résultats risque de ne pas être renvoyé au client ou demandeur pour l'une des raisons suivantes :

- La clause de renvoi des ensembles de résultats n'est pas spécifiée correctement dans la procédure fédérée.
- Certaines sources de données ne renvoient pas d'ensembles dans le même ordre à chaque appel d'une procédure. Etant donné que les procédures fédérées ne

renvoient que le premier ensemble de résultats, la source de données peut renvoyer un ensemble de résultats autre à l'appel de la procédure fédérée.

Par exemple, il existe deux procédures sur la source de données, PROCEDURE A et PROCEDURE B. PROCEDURE B appelle PROCEDURE A. Les instructions permettant de créer ces procédures sont les suivantes :

```
CREATE PROCEDURE A ()
BEGIN
  DECLARE cur1 CURSOR WITH RETURN TO CLIENT
  FOR SELECT * FROM t;
  OPEN cur1
END
CREATE PROCEDURE B (arg1 INT)
BEGIN
  DECLARE cur2 CURSOR WITH RETURN TO CLIENT
  FOR SELECT * FROM t;
  IF arg1<10) THEN
    CALL A();
  END IF;
  OPEN cur2
END;
```

La procédure fédérée FEDPROC1 référence PROCEDURE B de la source de données. L'instruction de la procédure FEDPROC1 est la suivante :

```
CREATE PROCEDURE FEDPROC1
SOURCE newton.B
FOR SERVER s1
NUMBER OF PARAMETERS 1
WITH RETURN TO CLIENT 1;
```

Une procédure locale appelle la procédure fédérée FEDPROC1. L'instruction de la procédure locale est la suivante :

```
CREATE PROCEDURE local (arg1 INT)
BEGIN
  CALL FEDPROC1 (arg1)
END;
```

Lorsque vous émettez l'instruction CALL LOCAL(1), l'ensemble de résultats cur1 de PROCEDURE A est renvoyé. L'ensemble de résultats cur2 n'est pas renvoyé.

Toutefois, si vous émettez l'instruction CALL LOCAL(20), l'ensemble de résultats cur2 de PROCEDURE B est renvoyé.

Session passe-système (Oracle uniquement)

Si vous créez une procédure, une fonction ou un module de source de données dans une session passe-système, un message de confirmant le succès de la création est renvoyé même si la définition d'objet contient une erreur. L'objet est créé sur le serveur Oracle, mais il affiche INVALID. Vous ne pouvez pas créer de procédures fédérées sur des objets INVALID. Lorsque vous essayez de créer une procédure fédérée qui référence un objet Oracle INVALID, l'instruction CREATE PROCEDURE (dérivée) échoue.

Utilisez l'une des méthodes suivantes pour déterminer la raison pour laquelle un objet n'est pas valide :

- Utilisez la commande SHOW ERRORS dans l'utilitaire SQL*Plus d'Oracle.
- Interrogez la table du catalogue sys.all_errors d'Oracle.

Chapitre 7. Création et modification de tables distantes à l'aide de DLL transparent

Avec le DLL transparent, vous pouvez utiliser les procédures que vous maîtrisez pour créer et modifier des tables distantes à l'aide de la base de données fédérée, sans utiliser de session passe-système.

Qu'est-ce que le DDL transparent ?

Le *DDL transparent* offre la possibilité de créer et de modifier des tables distantes par le biais de la base de données fédérée sans utiliser de sessions passe-système.

Les instructions SQL que vous utilisez avec le DDL transparent sont CREATE TABLE, ALTER TABLE et DROP TABLE.

Une instruction CREATE TABLE en DDL transparent crée une table distante sur la source de données et un pseudonyme pour cette table sur le serveur fédéré. Ainsi, les types de données DB2 que vous spécifiez sont mappés aux types de données distants à l'aide des mappages de type inversé par défaut. En général, les encapsuleurs assurent des mappages de type. Vous pouvez également créer des mappages inversés définis par l'utilisateur en remplacement des mappages par défaut.

L'avantage du DDL transparent est que les administrateurs de base de données peuvent utiliser des procédures qu'ils connaissent bien pour créer des tables tant locales que distantes. Le DDL transparent centralise l'administration des tables et facilite l'octroi d'autorisations.

Le DDL transparent est pris en charge avec les sources de données suivantes :

- DB2 for z/OS
- DB2 for System i
- DB2 Database pour Linux, UNIX et Windows
- DB2 Server for VM and VSE
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle
- Sybase
- Teradata

L'administrateur de base de données peut utiliser soit le Centre de contrôle DB2, soit des instructions DDL dans l'interpréteur de commandes DB2 pour créer les tables. L'utilisation du DDL transparent évite d'avoir à apprendre la syntaxe DDL différente requise pour chaque source de données.

Pour pouvoir créer des tables distantes sur une source de données par le biais de la base de données fédérée, vous devez configurer l'accès à la source de données :

- l'encapsuleur de cette source doit être enregistré dans le catalogue global ;

- la définition doit être créée pour le serveur à l'emplacement futur de la table distante ;
- des mappages utilisateur, si besoin est, doivent être créés entre le serveur fédéré et le serveur de source de données.

Utilisez l'assistant Création d'une table éloignée dans le Centre de contrôle DB2 pour créer des tables distantes.

L'ID autorisation des instructions DDL transparent doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- droit CREATETAB sur la base de données et privilège USE sur l'espace table, ainsi que l'un des suivants :
 - droit IMPLICIT_SCHEMA sur la base de données, si le nom de schéma implicite ou explicite de la table n'existe pas;
 - privilège CREATEIN sur le schéma, si le nom de schéma de la table fait référence à un schéma existant.

Pour émettre des instructions DDL transparent, votre ID autorisation doit bénéficier des droits d'accès nécessaires sur le pseudonyme (pour que le serveur fédéré accepte la demande), ainsi que les privilèges équivalents sur le serveur de source de données distant (pour que la source de données accepte la demande).

Colonnes LOB distantes et DDL transparent

Vous définissez la longueur d'une colonne LOB lors de l'utilisation de DDL transparent.

Certaines sources de données, comme Oracle et Informix, ne stockent pas les longueurs des colonnes LOB dans leurs catalogues système. Lorsque vous créez un pseudonyme sur une table, des informations provenant du catalogue système de la source de données sont extraites, dont la longueur de colonne. Comme aucune longueur n'est indiquée pour les colonnes LOB, la base de données fédérée suppose que cette longueur est égale à la longueur maximale d'une colonne LOB dans DB2 Database for Linux, UNIX and Windows. La base de données fédérée stocke la longueur maximale dans le catalogue de base de données fédérée comme longueur de la colonne de pseudonyme.

Cependant, lorsque vous créez une table distante à l'aide de DDL transparent, vous devez préciser la longueur de la colonne LOB. Lorsque le serveur fédéré crée un pseudonyme sur la table distante, il stocke indiquée dans le catalogue de la base de données fédérée comme longueur de la colonne de pseudonymes. La longueur maximale d'une colonne LOB est de 2 gigaoctets.

Création de tables distantes et de DDL transparent

Lorsqu'une table distante est créée par l'intermédiaire de la base de données fédérée à l'aide de DDL transparent, plusieurs opérations se produisent.

Lorsque vous créez la table distante :

- Un pseudonyme est créé automatiquement pour la table distante. Cet pseudonyme comporte le même nom que celui de la table spécifiée dans l'instruction CREATE TABLE. La table distante porte le même nom que la table, sauf si vous indiquez un autre nom à l'aide de l'option REMOTE_TABNAME.

- Le schéma de la table distante est le schéma du pseudonyme sauf si vous en indiquez un autre à l'aide de l'option `REMOTE_SCHEMA`.
- Le pseudonyme créé à l'aide de DDL transparent peut être utilisé comme tout autre pseudonyme. En outre, vous pouvez modifier (`ALTER`) et supprimer (`DROP`) la table distante (ce qui est impossible avec un pseudonyme créé à l'aide de `CREATE NICKNAME`).
- Une ligne est ajoutée dans la vue catalogue `SYSCAT.TABOPTIONS` avec le nom d'option `TRANSPARENT` et la valeur 'Y'.

Création de tables distantes à l'aide de DDL transparent

Pour créer une table distante à l'aide de DDL transparent, vous pouvez utiliser soit l'assistant du Centre de contrôle DB2, soit l'instruction `CREATE TABLE`.

Avant de commencer

Avant de créer une table distante, vous devez configurer le serveur fédéré pour accéder à cette source de données. Cette configuration comprend les opérations suivantes :

- création de l'encapsuleur pour ce type de source de données ;
- fourniture de la définition du serveur à l'emplacement futur de la table distante ;
- création des mappages utilisateur entre le serveur fédéré et le serveur de source de données.

Pour émettre des instructions DDL transparent, votre ID autorisation doit bénéficier des droits d'accès nécessaires sur le pseudonyme (pour que le serveur fédéré accepte la demande), ainsi que les privilèges équivalents sur le serveur de source de données distant (pour que la source de données accepte la demande).

L'ID autorisation qui émet les instructions DDL transparent doit détenir au moins l'un des privilèges suivants :

- droit `SYSADM` ou `DBADM`
- droit `CREATETAB` sur la base de données et privilège `USE` sur l'espace table, ainsi que l'un des suivants :
 - droit `IMPLICIT_SCHEMA` sur la base de données, si le nom de schéma implicite ou explicite de la table n'existe pas;
 - privilège `CREATEIN` sur le schéma, si le nom de schéma de la table fait référence à un schéma existant.

Restrictions

Les restrictions suivantes s'appliquent à la création d'une table distante à l'aide de DDL transparent :

- vous ne pouvez pas modifier ou supprimer les tables ayant été créées de façon native sur la source de données distante ;
- il est impossible de créer des tables de requêtes matérialisées sur des sources de données distantes ;
- vous pouvez indiquer des informations de colonne de base dans la définition de table, mais pas des options de table ou de colonne ; par exemple, les options `LOB` (`LOGGED` et `COMPACT`) ne sont pas prises en charge ;
- vous ne pouvez pas indiquer de commentaire sur une colonne ;
- vous ne pouvez pas générer le contenu des colonnes ;

- vous pouvez définir une clé primaire, mais pas une clé externe ni des contraintes de vérification ; les colonnes utilisées pour une clé primaire doivent être NOT NULL et ne peuvent pas inclure d'objets LOB ;
- vous ne pouvez pas modifier les paramètres des colonnes existantes, telles que le type ou la longueur des données ;
- la clause DEFAULT de l'instruction CREATE TABLE n'est pas prise en charge.

A propos de cette tâche

Utilisez l'assistant de table distante dans le Centre de contrôle DB2 pour éviter d'indiquer un paramètre ou une option qui ne soit pas pris en charge. L'assistant permet de spécifier des colonnes en les sélectionnant dans une liste de colonnes prédéfinies ou en indiquant les attributs d'une nouvelle colonne.

Procédure

Pour créer une table distante à partir de l'invite de la ligne de commande, émettez l'instruction CREATE TABLE en définissant les paramètres appropriés.

Pour créer une table distante dans le Centre de contrôle DB2, utilisez l'assistant Création d'une table éloignée.

1. Développez le dossier **Objets de base de données fédérée**.
2. Développez les objets encapsuleur et définition de serveur à partir de la source de données pour laquelle vous souhaitez créer une table distante.
3. Cliquez avec le bouton droit de la souris sur le dossier **Tables éloignées**, puis cliquez sur **Création**. L'assistant Création d'une table éloignée est lancé.
4. Effectuez les étapes de cet assistant.

Création de tables distantes à l'aide de DDL transparent - exemples

Les exemples suivants illustrent la création de tables distantes à l'aide de DDL transparent et l'utilisation de mappages de types de données.

Lorsque vous créez des tables distantes à l'aide de DDL transparent :

- la source de données distante doit prendre en charge les types de données de colonne et l'option de clé primaire de l'instruction CREATE TABLE ;
Exemple : si la source de données distante ne prend pas en charge les clés primaires, une erreur risque d'être renvoyée ou la demande d'être ignorée, selon la réponse de la source de données aux demandes qu'elle ne prend pas en charge ;
- le serveur distant doit être indiqué dans la clause OPTIONS clause. Cette clause peut être utilisée pour remplacer le nom distant ou le schéma distant de la table en cours de création. L'option SQL_SUFFIX est autorisée à la fin de l'instruction CREATE TABLE. Vous pouvez indiquer cette option pour une source de données relationnelle afin d'ajouter des options propres à la source de données à l'instruction CREATE TABLE qui est émise sur cette source.

Exemple : Vous souhaitez créer la table EMPLOY sur un serveur Oracle. Dans l’instruction CREATE TABLE, utilisez les types de données DB2 lorsque vous spécifiez chaque colonne. Dans l’interpréteur de commandes, la syntaxe permettant de créer la table est la suivante :

```
CREATE TABLE EMPLOY
( EMP_NO CHAR(6) NOT NULL,
  FIRSTNAME VARCHAR(12) NOT NULL,
  MIDINT CHAR(1) NOT NULL,
  LASTNAME VARCHAR(15) NOT NULL,
  HIREDATE DATE,
  JOB CHAR(8),
  SALARY DECIMAL(9,2),
  PRIMARY KEY (EMP_NO) )
OPTIONS (REMOTE_SERVER 'ORASERVER',
        REMOTE_SCHEMA 'J15USER1', REMOTE_TABNAME 'EMPLOY' )
```

EMPLOY

Nom du pseudonyme associé à la table.

REMOTE_SERVER 'ORASERVER'

Nom que vous avez fourni pour le serveur dans l’instruction CREATE SERVER. Cette valeur fait la distinction majuscules/minuscules.

REMOTE_SCHEMA 'J15USER1'

Nom du schéma distant. Bien que ce paramètre soit facultatif, il est recommandé d’indiquer un nom de schéma. Si ce paramètre n’est pas indiqué, le schéma de pseudonyme est utilisé pour le nom de schéma distant. Cette valeur fait la distinction majuscules/minuscules.

REMOTE_TABNAME 'EMPLOY'

Nom de la table distante. Ce paramètre est facultatif. S’il n’est pas indiqué, le nom de table local est utilisé pour le nom de table distante. Cette valeur doit être un nom valide sur la source de données distante et ne peut pas désigner un nom de table existant. Cette valeur fait la distinction majuscules/minuscules.

Dans l’exemple ci-dessus, la base de données fédérée utilise des mappages de types de données inversés pour mapper les types de données DB2 sur des types de données Oracle. Sur le serveur Oracle distant, la table EMPLOY est créée à l’aide de types de données Oracle. Le tableau suivant donne les mappages des types de données DB2 sur des types de données Oracle pour les colonnes indiquées dans l’exemple.

Tableau 7. Exemple de mappages de types de données inversés entre la base de données fédérée et Oracle

Colonne	Type de données DB2 indiqué dans l’instruction CREATE TABLE	Type de données Oracle utilisé dans la table distante
EMP_NO	CHAR(6) NOT NULL	CHAR(6) NOT NULL
FIRST_NAME	VARCHAR(12) NOT NULL	VARCHAR2(12) NOT NULL
MID_INT	CHAR(1) NOT NULL	CHAR(1) NOT NULL
LAST_NAME	VARCHAR(15) NOT NULL	VARCHAR2(15) NOT NULL
HIRE_DATE	DATE	DATE
JOB	CHAR(8)	CHAR(8)
SALARY	DECIMAL(9,2)	NUMBER(9,2)

Modification de tables distantes à l'aide de DDL transparent

Vous pouvez modifier des tables de source de données distantes créées par l'intermédiaire de la base de données fédérée à l'aide de DDL transparent. Vous ne pouvez pas modifier les tables ayant été créées directement sur la source de données distante.

Avant de commencer

L'ID autorisation des instructions DDL transparent doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- droit CREATETAB sur la base de données et privilège USE sur l'espace table, ainsi que l'un des suivants :
 - droit IMPLICIT_SCHEMA sur la base de données, si le nom de schéma implicite ou explicite de la table n'existe pas;
 - privilège CREATEIN sur le schéma, si le nom de schéma de la table fait référence à un schéma existant.

Pour émettre des instructions DDL transparent, votre ID autorisation doit bénéficier des droits d'accès nécessaires sur le pseudonyme (pour que le serveur fédéré accepte la demande), ainsi que les privilèges équivalents sur le serveur de source de données distant (pour que la source de données accepte la demande).

Restrictions

Les restrictions suivantes s'appliquent à la modification d'une table distante à l'aide de DDL transparent :

- vous ne pouvez pas modifier les tables ayant été créées de façon native sur la source de données distante ;
- une clé primaire existante ne peut pas être modifiée ou supprimée dans une table distante ;
- La modification d'une table distante invalide les modules qui dépendent du pseudonyme associé à cette table ;
- la source de données distante doit prendre en charge les modifications apportées par l'instruction ALTER TABLE ; par exemple, si la source de données distante ne prend pas en charge les clés primaires, une erreur risque d'être renvoyée ou la demande d'être ignorée, selon la réponse de la source de données aux demandes qu'elle ne prend pas en charge ;
- vous ne pouvez pas indiquer de commentaire sur une colonne ;
- vous ne pouvez pas générer le contenu des colonnes ;
- vous pouvez définir une clé primaire, mais pas une clé externe ni des contraintes de vérification ; les colonnes utilisées pour une clé primaire doivent être NOT NULL et ne peuvent pas inclure d'objets LOB ;
- vous ne pouvez pas modifier les paramètres des colonnes existantes, telles que le type ou la longueur des données ;
- la clause DEFAULT de l'instruction ALTER TABLE n'est pas prise en charge.

A propos de cette tâche

Vous pouvez utiliser soit le Centre de contrôle DB2, soit l'instruction ALTER TABLE pour modifier les tables créées par le biais de IBM InfoSphere Federation Server à l'aide de DDL transparent. Utilisez le Centre de contrôle DB2 pour éviter d'indiquer un paramètre ou une option qui ne soit pas pris en charge. L'instruction ALTER TABLE permet d'effectuer les opérations suivantes :

- ajouter de nouvelles colonnes ;
- ajouter la clé primaire de la table.

N'utilisez pas l'instruction ALTER TABLE pour ajouter ou modifier des options de colonne. Servez-vous plutôt de l'instruction ALTER NICKNAME.

Procédure

Pour modifier une table distante à l'aide de DDL transparent, émettez l'instruction ALTER TABLE :

Exemple : Vous souhaitez ajouter une clé primaire sur une table distante EMPLOYEE que vous avez créée à l'aide de DDL transparent. Utilisez l'instruction ALTER TABLE pour modifier la table :

```
ALTER TABLE EMPLOYEE
  ADD PRIMARY KEY (EMP_NO, WORK_DEPT)
```

Les colonnes utilisées pour une clé primaire doivent être NOT NULL et ne peuvent pas inclure d'objets LOB.

Exemple : Vous souhaitez ajouter les colonnes ORDER_DATE et SHIP_DATE à la table distante SPALTEN créée à l'aide de DDL transparent. Utilisez l'instruction ALTER TABLE suivante pour créer la table :

```
ALTER TABLE SPALTEN
  ADD COLUMN ORDER_DATE DATE
  ADD COLUMN SHIP_DATE DATE
```

Suppression de tables distantes à l'aide de DDL transparent

Vous pouvez supprimer des tables de source de données distantes créées par l'intermédiaire de la base de données fédérée à l'aide de DDL transparent. Vous ne pouvez pas supprimer les tables ayant été créées directement sur la source de données distante.

Avant de commencer

L'ID autorisation des instructions DDL transparent doit détenir au moins l'un des privilèges suivants :

- droit SYSADM ou DBADM
- droit CREATETAB sur la base de données et privilège USE sur l'espace table, ainsi que l'un des suivants :
 - droit IMPLICIT_SCHEMA sur la base de données, si le nom de schéma implicite ou explicite de la table n'existe pas;
 - privilège CREATEIN sur le schéma, si le nom de schéma de la table fait référence à un schéma existant.

Pour émettre des instructions DDL transparent, votre ID autorisation doit bénéficier des droits d'accès nécessaires sur le pseudonyme (pour que le serveur fédéré accepte la demande), ainsi que les privilèges équivalents sur le serveur de source de données distant (pour que la source de données accepte la demande).

Restrictions

Vous ne pouvez pas supprimer les tables ayant été créées de façon native sur la source de données distante.

A propos de cette tâche

Pour supprimer une table distante créée par l'intermédiaire de la base de données fédérée à l'aide de DDL transparent, vous pouvez utiliser soit le Centre de contrôle DB2, soit l'instruction DROP.

La suppression d'un pseudonyme pour une table distante créé à l'aide de DDL transparent entraîne seulement la suppression du pseudonyme local de cette table. En effet, l'instruction DROP NICKNAME ne supprime pas la table distante. Vous devez pour cela utiliser l'instruction DROP TABLE.

La suppression d'une table distante entraîne d'abord la suppression de cette table sur la source de données, puis la suppression du pseudonyme correspondant de la table distante dans la base de données fédérée. La suppression du pseudonyme entraîne l'invalidation des modules basés sur ce pseudonyme.

Procédure

Pour supprimer une table distante, émettez l'instruction DROP TABLE.

Exemple : Pour supprimer une table appelée SPALTEN, émettez l'instruction DROP suivante :

```
DROP TABLE SPALTEN
```

où *SPALTEN* est le nom local de la table distante.

Chapitre 8. Gestion de transactions dans un système fédéré

Le traitement de transactions d'un système fédéré vous permet de lire et de mettre à jour des bases de données dans une seule transaction, tout en préservant la cohérence des données. Les systèmes fédérés supportent les protocoles de validation en une et deux phases. Lorsque vous administrez votre système fédéré, vous devez configurer le protocole approprié.

Description du support des transactions du système fédéré

Le fait de connaître les concepts de traitement des transactions dans un environnement distribué DB2 Database for Linux, UNIX et Windows permet de mieux comprendre les transactions du système fédéré.

Pour comprendre le traitement des transactions du système fédéré, vous devez connaître les concepts de traitement des transactions réparties suivants :

- Unité d'oeuvre (UOW)
- Unité d'oeuvre éloignée (RUOW)
- Unité d'oeuvre répartie (DUOW)
- Mise à jour multisite
- Gestionnaire de transactions (TM)
- Gestionnaire de ressources (RM)
- Connexion de type 1
- Connexion de type 2
- Validation en une phase
- Validation en deux phases

Ces concepts fonctionnent de la même manière dans les systèmes de base de données fédérés et non fédérés. Toutefois, la portée de chaque concept change dans un système fédéré.

Par exemple, une unité d'oeuvre commence de manière implicite lorsque des données de la base de données sont lues ou écrites. Pour une unité d'oeuvre d'un système fédéré, la base de données peut être une base de données fédérée ou une base de données de source de données. Pour une unité d'oeuvre répartie d'un système fédéré, vous pouvez accéder à la fois à une base de données fédérée et à une base de données de sources de données.

Une application doit arrêter une unité d'oeuvre en émettant une instruction COMMIT ou ROLLBACK, quel que soit le nombre de bases de données auxquelles vous accédez. L'instruction COMMIT rend permanentes toutes les modifications effectuées dans une unité d'oeuvre. L'instruction ROLLBACK supprime ces modifications d'une base de données. Les modifications effectuées par une unité d'oeuvre deviennent visibles pour les autres utilisateur une fois la validation effectuée.

Recommandation : Validez ou annulez de manière explicite des unités d'oeuvre dans vos applications.

Dans une unité d'oeuvre répartie qui implique les mises à jour de plusieurs bases de données sur plusieurs sites, les données doivent être cohérentes. Le protocole de validation en deux phases ou de mise à jour multisite est généralement utilisé pour vérifier la cohérence des données entre plusieurs bases de données d'une unité d'oeuvre.

Les transactions fédérées prennent en charge les protocoles de validation en une phase et en deux phases. L'option de serveur DB2_TWO_PHASE_COMMIT active le support de validation en deux phases pour les sources de données suivantes :

- Sources de données de la famille DB2
- Informix
- Oracle
- Sybase
- MS SQL Server

Lorsqu'une source de données est déclarée en tant que source de données de validation en deux phases fédérées, autrement dit l'option de serveur DB2_TWO_PHASE_COMMIT, a la valeur «Y», une validation pour cette source de données utilise ce protocole de validation en deux phases même s'il s'agit d'une transaction de mise à jour d'un seul site ou d'une transaction de mise à jour de plusieurs sites.

Lorsqu'une source de données est déclarée en tant que source de données de validation en une phase fédérée (valeur par défaut) et qu'il s'agit d'une transaction de mise à jour d'un seul site, une validation pour cette source de données utilise le protocole de validation en une phase.

Dans l'exemple suivant d'opération de validation en une phase, Oracle est défini en tant que source de données de validation en une phase :

```
SELECT *  
FROM pseudonyme_oracle  
UPDATE pseudonyme_oracle  
COMMIT
```

Dans l'exemple suivant d'une opération de validation en deux phases, Oracle et DRDA sont définis en tant que sources de données en deux phases :

```
SELECT *  
FROM pseudonyme_oracle  
UPDATE pseudonyme_oracle  
  
SELECT * FROM pseudonyme_drda  
UPDATE pseudonyme_drda  
COMMIT
```

Mise à jour dans un système fédéré

Dans un système fédéré, une mise à jour n'est pas juste une transaction qui inclut une instruction INSERT, UPDATE ou DELETE. Certaines opérations sont considérées comme des mises à jour dans un système fédéré et certains types de mise à jour sont admis dans un un système fédéré.

Dans un système fédéré, les mises à jour peuvent être effectuées localement ou à distance.

- Les mises à jour de site local sont des mises à jour de tables ou de vues DB2 qui ne référencent pas les pseudonymes
- Les mises à jour éloignées sont des mises à jour d'objets sur une source de données éloignée. Les sources de données éloignées incluent :
 - Une autre base de données ou instance DB2 Database for Linux, UNIX et Windows sur le serveur fédéré
 - Une autre base de données ou instance DB2 Database pour Linux, UNIX et Windows sur un autre serveur
 - Des sources de données autres que DB2 Database for Linux, UNIX et Windows, comme DB2 for System i, Informix, Oracle et Teradata

Il existe quatre types d'action que le serveur fédéré considère comme des transactions de mise à jour. Le tableau suivant décrit les mises à jour que vous pouvez effectuer sur un système fédéré.

Tableau 8. Types de mise à jour et site sur lequel les mises à jour sont effectuées

Type d'action	Site local	Site éloigné	Explication
Mise à jour locale (DDL et DML)	Y	N	Mise à jour sur un objet dans la base de données fédérée.
Mise à jour éloignée (pseudonyme)	N	Y	Mise à jour sur un objet de source de données éloigné pour lequel vous avez créé un pseudonyme.
SQL dynamique dans des sessions passe-système	N	Y	Mise à jour sur un objet de source de données éloigné. Vous ne pouvez pas utiliser de session passe-système pour mettre à jour des objets locaux. Même les requêtes SELECT envoyées dans des sessions passe-système sont considérées comme action de mise à jour.
Transparent DDL	Y	Y	Paire de transactions qui crée, modifie ou supprime des tables et leurs pseudonymes correspondants dans une base de données fédérée. Par exemple, une paire de transactions qui crée une table éloignée sur une source de données et un pseudonyme sur le serveur fédéré.

Transaction de mise à jour dans une session passe-système

Un serveur fédéré traite toutes les instructions SQL dynamiques envoyées via des sessions passe-système en tant que mises à jour. Ce comportement garantit l'intégrité des données.

Si une instruction SQL dynamique envoyée via une session passe-système aboutit, la transaction est enregistrée en tant que mise à jour. L'instruction SQL peut être de tout type, elle peut être notamment une instruction SELECT.

Source de données validant de manière automatique des instructions DDL

Certaines sources de données, telles Oracle, valident automatiquement la transaction en cours sur les sites de source de données lors de l'exécution d'une instruction DDL.

Si vous créez une table éloignée à l'aide d'éléments DDL transparents ou dans une session passe-système, ces sources de données ne peuvent pas annuler la table éloignée une fois la table créée. Vous devez supprimer la table éloignée manuellement.

Fonctions définies par l'utilisateur transmises à la source de données pour traitement

Si une fonction définie par l'utilisateur effectue une mise à jour sur une source de données, le serveur fédéré n'est pas notifié de la mise à jour.

Etant donné que le serveur fédéré ne traite pas ces fonctions définies par l'utilisateur en tant qu'instructions de mise à jour, l'intégrité de la protection au niveau des instructions appliquée par le système fédéré aux opérations de mise à jour n'est pas applicable. Par conséquent, l'intégrité des données peut être compromise dans certaines situations.

Important : L'intégrité des données ne peut pas être garantie lorsqu'une fonction définie par l'utilisateur transmise à une source de données effectue une mise à jour.

Chapitre 9. Transactions de validation en deux phases

Vous pouvez utiliser la fonction de validation en deux phases d'un système fédéré pour mettre à jour les données d'une ou plusieurs sources dans une même transaction.

Validation en deux phases pour les transactions fédérées

Un système fédéré peut utiliser la validation en deux phases pour les transactions qui accèdent à une ou plusieurs sources de données. La validation en deux phases utilise le protocole X/Open XA de norme de l'industrie afin de coordonner le traitement des transactions d'unité d'oeuvre répartie.

Dans une opération de validation en deux phases, le traitement de validation se produit en deux phases : phase de préparation et phase de validation. Lors de la phase de préparation dans un système fédéré, un serveur fédéré interroge l'ensemble des sources de données de la validation en deux phases impliquées dans une transaction. Cette activité d'interrogation vérifie si chaque source de données est prête pour la validation ou l'annulation des données. Lors de la phase de validation, le serveur fédéré indique à chaque source de données de validation en deux phases de valider les données ou d'annuler la transaction.

Dans un environnement de validation en une phase, plusieurs sources de données sont mises à jour l'une après l'autre à l'aide d'opérations de validation séparées. Des incidents de synchronisation peuvent survenir lorsque certaines sources de données sont mises à jour avec succès et d'autres ne le sont pas.

Par exemple, si une transaction supprime des fonds d'un compte et les place dans un autre compte à l'aide d'une validation en une phase, le système peut avec succès valider l'opération de retrait et ne pas mener à terme l'opération de dépôt. L'opération de dépôt peut être annulée contrairement à l'opération de retrait car sa validation a déjà abouti. Par conséquent, les fonds seront virtuellement "perdus".

Dans un environnement de validation en deux phases, les transactions de retrait et de dépôt sont préparées ensemble et validées ou annulées ensemble. Par conséquent, l'intégrité des montants des fonds reste intacte.

Planification pour la validation en deux phases fédérées

La validation en deux phases fédérées n'offre pas d'avantages dans tous les environnements commerciaux. Il existe plusieurs facteurs à prendre en compte avant de décider de déployer la validation en deux phases fédérées.

Pour utiliser la validation en deux phases pour les transactions fédérées, prenez en compte les points suivants :

- Votre système d'exploitation et l'environnement de source de données peuvent-ils prendre en charge la validation en deux phases pour les transactions fédérées ?
- Votre environnement commercial requiert-il la validation en deux phases pour les transactions fédérées ?

Pour décider si la validation en deux phases est adaptée à votre environnement commercial, vous devez comprendre comment fonctionne la validation en deux phases pour les transactions fédérées ainsi que les incidents résolus.

- Mode de configuration du serveur fédéré et des sources de données compatibles pour l'utilisation de la validation en deux phases des transactions fédérées.
Il existe des exigences de source de données et de serveur fédéré de base pour l'utilisation de la validation en deux phases des transactions fédérées. Vous devez également prendre en compte les performances lors du déploiement de la validation en deux phases.
- Pour résoudre manuellement les transactions en attente de validation, vous devez connaître le fonctionnement interne de la validation en deux phases pour les transactions fédérées.

La validation en deux phases des transactions fédérées peut résoudre les incidents sans intervention. En cas d'indisponibilité du réseau, d'erreurs matérielles ou d'une nécessité urgente de libération de ressources, vous pouvez manuellement résoudre les incidents via le *traitement heuristique*.

Architecture fédérée pour la validation en deux phases

La validation en deux phases fédérées dépend de la fonction de validation en deux phases disponible dans DB2. Dans la validation en deux phases, le modèle DTP (Distributed Transaction Processing) X/Open a plusieurs composants : identificateurs de transaction, gestionnaires de transaction et gestionnaires de ressources. Dans des systèmes fédérés utilisant la validation en deux phases fédérées, un autre composant est ajouté, le gestionnaire de transactions fédéré.

Un serveur fédéré devient un gestionnaire de transactions fédérées lorsque le serveur coordonne l'activité pour une ou plusieurs sources de données éloignées qui utilisent le protocole de validation en deux phases. Un gestionnaire de transactions fédérées effectue des fonctions de gestion de transactions au nom du gestionnaire de transactions. Le client ou l'application qui initie une transaction d'unité d'oeuvre répartie et le gestionnaire de transactions ne sont pas notifiés de l'activité que le gestionnaire de transactions fédérées coordonne sur les sources de données éloignées. Le gestionnaire de transactions fédéré communique avec les gestionnaires de transactions de la base de données DB2 via une interface XA. Outre les exigences de X/Open pour la validation en deux phases, la base de données du gestionnaire de transactions doit être accessible à partir de l'instance fédérée. Les gestionnaires de ressources suivent les instructions fournies par un gestionnaire de transactions fédérées pour la validation ou l'annulation d'une transaction.

La figure suivante présente un exemple de transaction de validation en deux phases dans un système fédéré standard à partir du lancement client jusqu'aux mises à jour de la source de données.

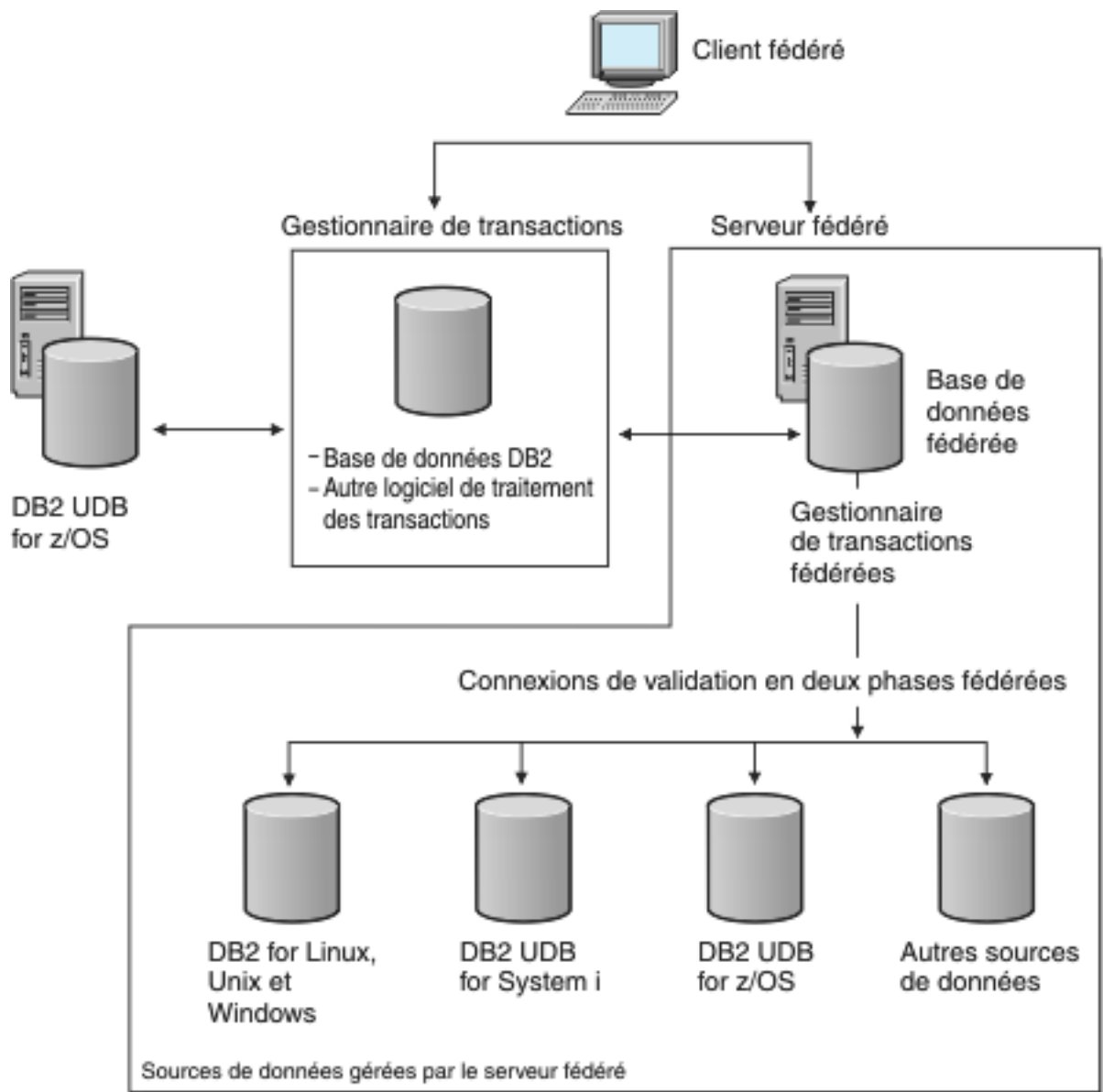


Figure 3. Transaction de validation en deux phases fédérées simple

Dans la figure précédente, la connexion du client au gestionnaire de transactions est une connexion de type 2. Chaque connexion de base de données dispose également de son propre paramètre de point de synchronisation. Un point de synchronisation correspond à un moment où toutes les données récupérables auxquelles un programme accède sont cohérentes. Les connexions à deux phases de point de synchronisation prennent en charge les transactions d'unité d'oeuvre répartie avec des mises à jour sur plusieurs sources de données.

Lorsque le client se connecte à la base de données DB2, le gestionnaire de transactions n'est pas notifié de la transaction mais aucune coordination supplémentaire n'est requise du serveur fédéré. Lorsque le serveur fédéré se connecte aux sources de données à l'aide du protocole de validation en deux phases, le serveur fédéré devient le gestionnaire de transactions fédérées. Le serveur fédéré contrôle et coordonne les validations en deux phases. Le gestionnaire de transactions n'est pas alors notifié des transactions de validation en deux phases dans les sources de données. Le gestionnaire de transactions sait uniquement qu'une seule transaction est traitée avec le serveur fédéré.

Les sources de données ne peuvent pas initier la resynchronisation lorsqu'une erreur se produit dans un système fédéré. Le serveur fédéré initie le processus de resynchronisation.

Les résultats peuvent être imprévisibles si vous tentez d'accéder à une source de données en utilisant plusieurs chemins dans la même transaction avec la validation en deux phases fédérées. Par exemple, si le serveur fédéré est un gestionnaire de ressources pour un gestionnaire de transactions externe, il est possible d'accéder à la source de données de manière indirecte à partir du serveur fédéré et de manière directe en tant que gestionnaire de ressources du gestionnaire de transactions. Dans ce cas, la source de données peut ne pas indiquer si ces deux chemins proviennent de la transaction globale. La source de données peut créer deux entrées de transaction pour la même transaction globale et traite chaque transaction en tant qu'élément séparé, ce qui peut générer des résultats imprévisibles. La source de données peut également détecter que deux chemins proviennent de la même transaction globale et rejette le deuxième chemin.

Validation en deux phases pour les transactions fédérées - Exemples

Un système fédéré qui utilise la validation en deux phases peut être configuré de différentes manières. Les choix de configuration dépendent de la solution requise.

Les configurations peuvent utiliser des connexions de type 1 ou de type 2.

Les connexions de type 1 sont des connexions dans lesquelles un processus d'application est connecté à un serveur d'applications conformément aux règles de l'unité d'oeuvre.

Les connexions de type 2 sont des connexions dans lesquelles un processus d'application est connecté à un serveur d'applications et établit les règles pour l'unité d'oeuvre répartie gérée par les applications. Le serveur d'applications devient le serveur du processus.

La figure suivante affiche une connexion DB2 de type 1 avec un serveur fédéré fonctionnant en tant que gestionnaire de transactions.

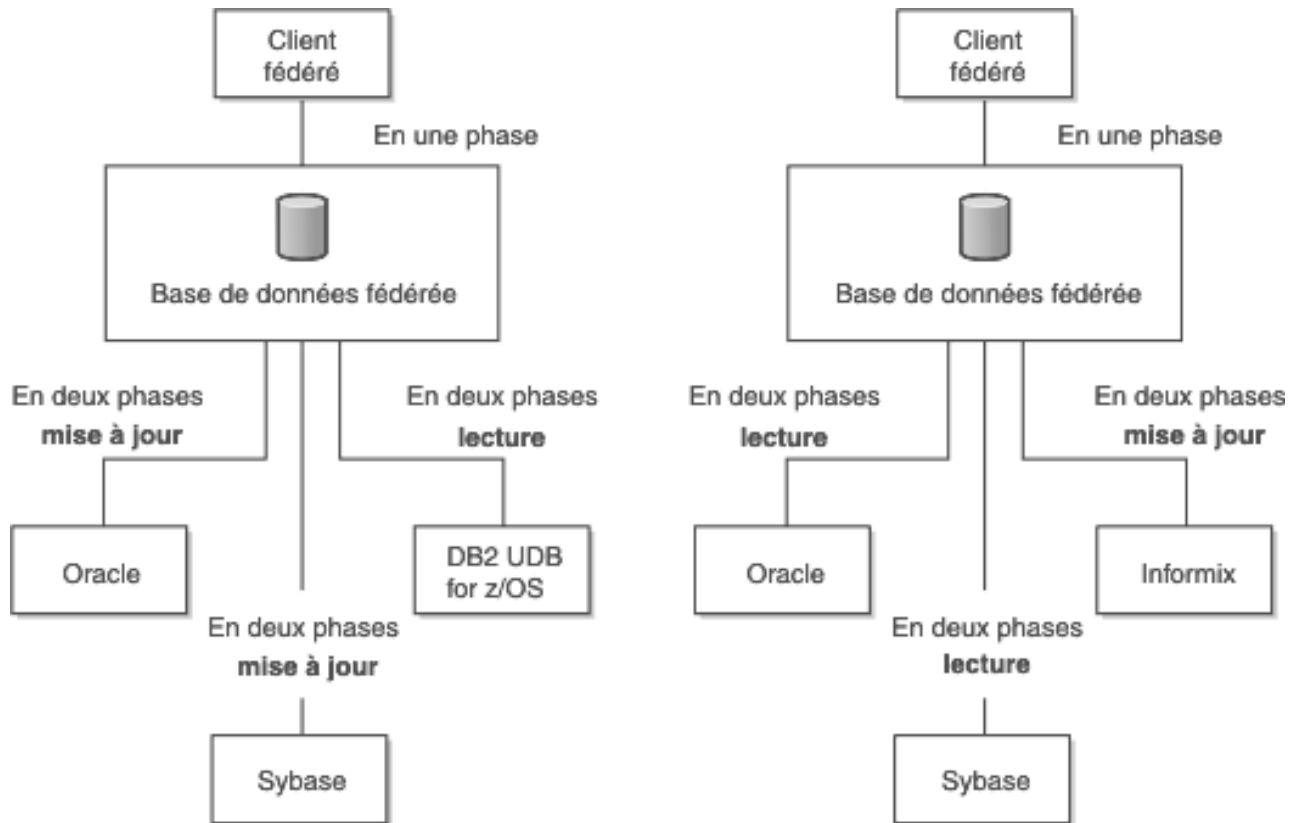


Figure 4. Connexion DB2 de type 1 avec un serveur fédéré en tant que gestionnaire de transactions

La figure suivante affiche une connexion DB2 de type 2 avec un serveur fédéré fonctionnant en tant que gestionnaire de ressources. Dans cette configuration, toutes les sources de données fédérées doivent être prises en charge pour la validation en deux phases et activées pour la validation en deux phases fédérées.

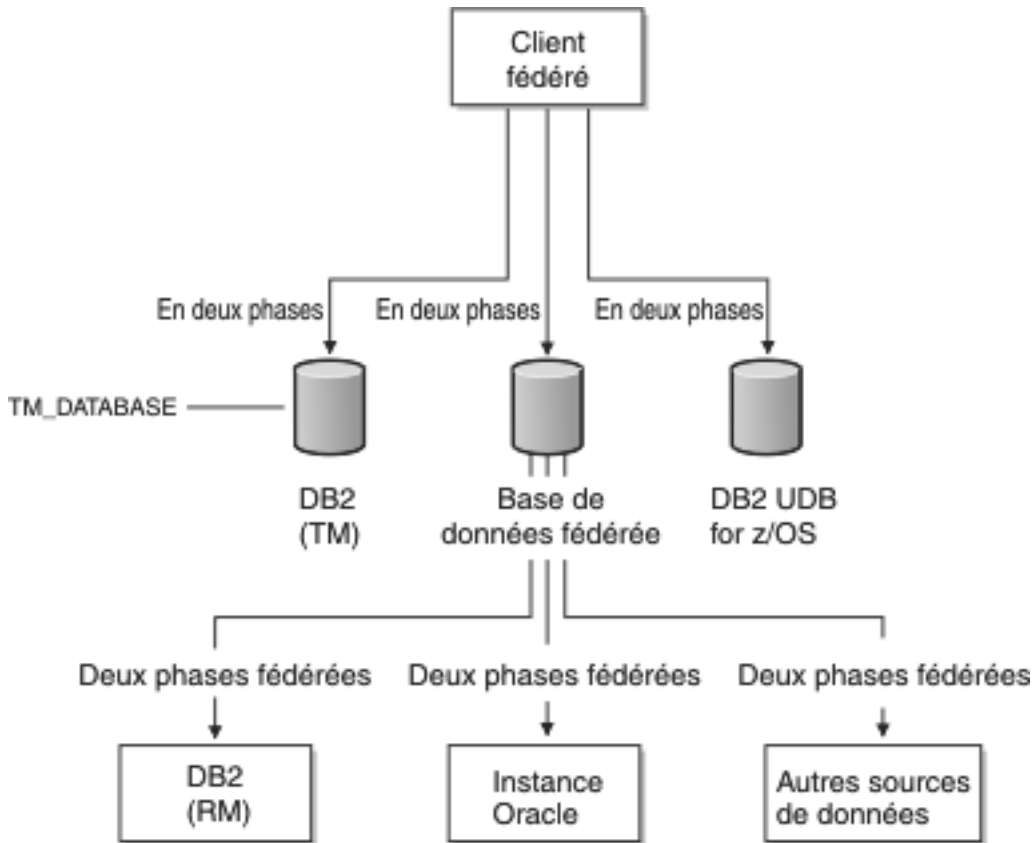


Figure 5. Connexion DB2 de type 2 avec le serveur fédéré en tant que gestionnaire de ressources

La figure suivante affiche une connexion DB2 de type 2 avec un serveur fédéré fonctionnant en tant que gestionnaire de transactions.

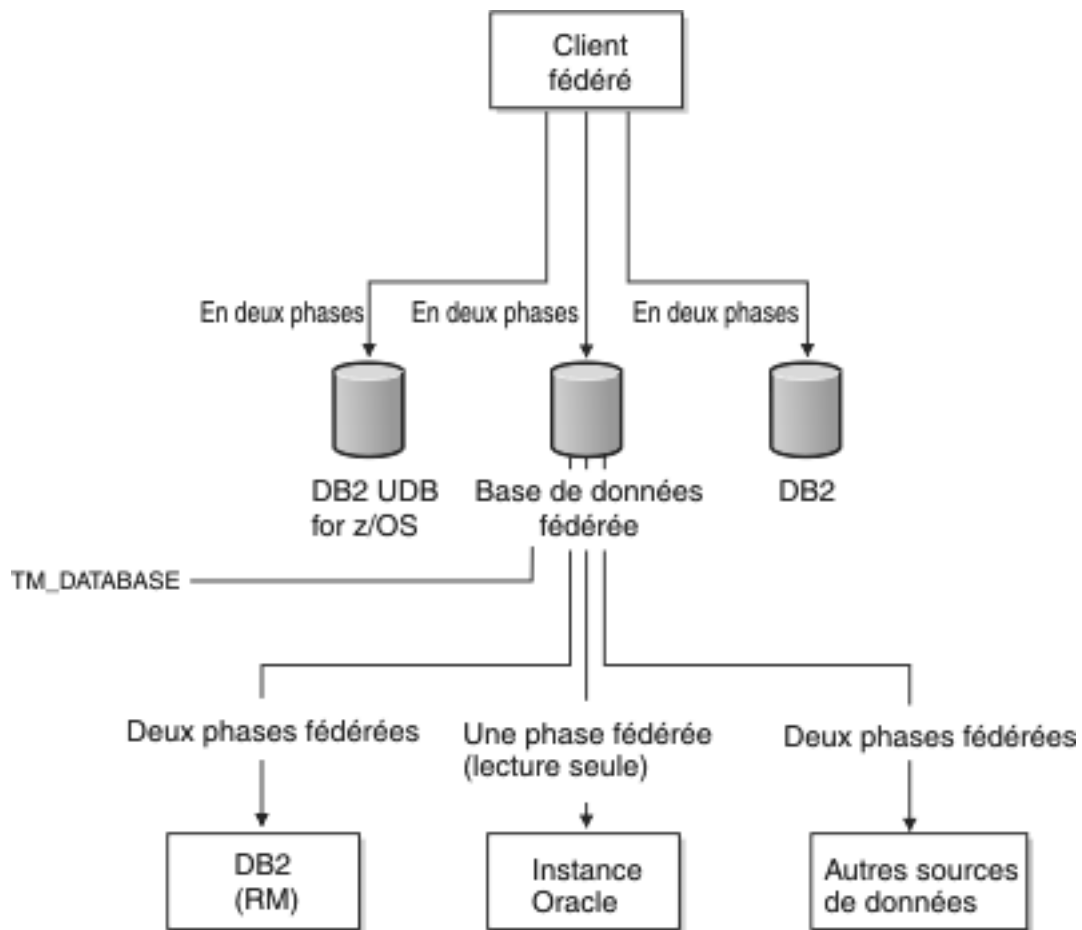


Figure 6. Connexion DB2 de type 2 avec un serveur fédéré en tant que gestionnaire de transactions

La figure suivante affiche une connexion XA avec un serveur fédéré fonctionnant comme gestionnaire de ressources.

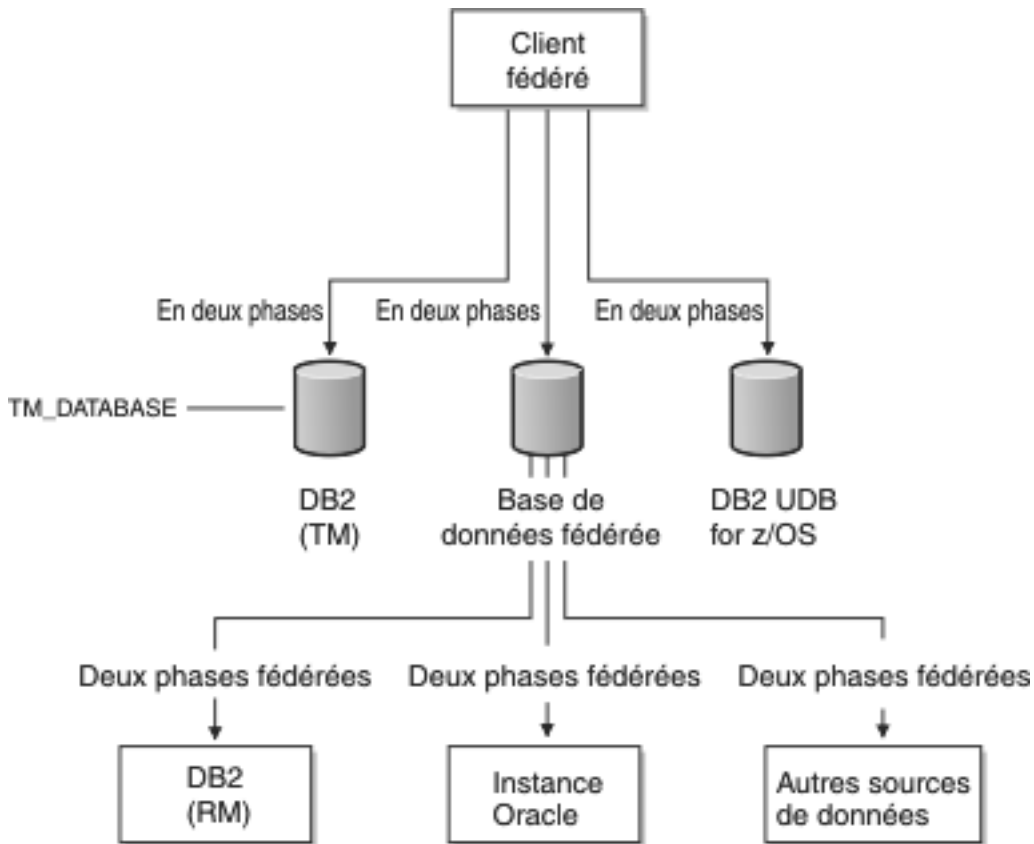


Figure 7. Connexion XA avec le serveur fédéré en tant que gestionnaire de ressources

Procédure de traitement des transactions de validation en deux phases

Le serveur fédéré conserve l'atomicité et la cohérence des données des sources de données qu'il gère. La gamme des transactions possibles dépend du type de connexion. Elle varie lorsque le serveur fédéré est le gestionnaire de transactions ou le gestionnaire de ressources pour la connexion.

L'atomicité un principal de base de données dans lequel des ensembles d'opérations sont définis dans des transactions ne pouvant pas être fractionnés. Ce principal garantit que la base de données est constamment cohérente car si une opération de la transaction n'aboutit pas, l'intégralité de la transaction est en situation d'échec au lieu de la compromission de l'intégrité des données suite à un changement partiel.

Par exemple, une transaction permettant de transférer des fonds d'un compte à un autre implique le retrait de fonds du premier compte et l'ajout de fonds au deuxième compte. Si seul le retrait aboutit, les fonds cessent d'exister dans le premier compte.

Les processus du serveur fédéré mettent à jour les requêtes selon des règles strictes. Une mise à jour fédérée correspond à une des actions suivantes :

- Opération d'insertion, de mise à jour ou de suppression fédérée où la source de données correspondante prend en charge les opérations d'insertion, de mise à jour ou de suppression. Par exemple, certaines sources de données ne prennent pas en charge des opérations de mise à jour. Certaines sources de données sont en lecture seule, dans ce cas les opérations d'insertion, de mise à jour ou de suppression fédérées ne sont pas admises.
- Opération passe-système réussie dans une session passe-système.
- Opération DDL transparent, qui est considérée à la fois comme une mise à jour locale et une mise à jour fédérée car elle effectue des opérations de mise à jour de base de données localement et à distance.
- Procédure mémorisée fédérée avec MODIFY SQL ACCESS.

Remarque : N'utilisez pas plusieurs chemins pour accéder à la même source de données dans une seule transaction. Une telle transaction peut se bloquer. Autrement dit, la transaction peut être interrompue de manière inattendue. Par exemple, n'utilisez pas plusieurs serveurs fédérés afin de faire référence aux mêmes sources de données dans la même transaction.

Le tableau suivant répertorie les événements survenant dans une transaction d'unité d'oeuvre répartie incluant le type de connexion, le type de validation, le rôle du serveur fédéré dans la transaction, les opérations admises et le mode d'utilisation de l'élément DDL transparent.

Tableau 9. Événements qui surviennent dans une transaction d'unité d'oeuvre répartie

Type de validation	Type de connexion	Rôle du serveur fédéré	Opérations	Transparent DDL
En une phase	Transaction locale DB2 de type 1 ou XA	Gestionnaire de sous-transactions. Se comporte comme le coordinateur de transactions, détermine la sortie des transactions et la transmet à chaque gestionnaire de ressources participant.	Les opérations de lecture de validation en une phase et en deux phases sont admises. Une source de données en une phase peut être mise à jour tant qu'il s'agit de la seule mise à jour de la transaction.	Éléments admis et gérés en fonction de règles de source de données de validation en une phase. Chaque instruction émise doit être la seule mise à jour dans une transaction de validation en une phase. Ne peut pas exister avec d'autres mises à jour de source de données de validation en deux phases dans la même transaction. Il est fortement recommandé que les instructions COMMIT ou ROLLBACK soient émises avant et après les transactions DDL transparentes.

Tableau 9. Événements qui surviennent dans une transaction d'unité d'oeuvre répartie (suite)

Type de validation	Type de connexion	Rôle du serveur fédéré	Opérations	Transparent DDL
En deux phases	Transaction locale DB2 de type 1 ou XA	Gestionnaire de transactions. Se comporte comme le coordinateur de transactions, détermine la sortie des transactions et la transmet à chaque gestionnaire de ressources participant.	Les opérations de lecture de validation en une phase et en deux phases sont admises. Plusieurs sources de données en deux phases peuvent être mises à jour.	Éléments admis et gérés en fonction des règles de source de données de validation en deux phases. Peut exister avec d'autres mises à jour de source de données de validation en une phase ou en deux phases dans la même transaction.
En une phase	Transaction globale DB2 de type 2 ou XA	Il peut s'agir du gestionnaire de transactions. Si ce n'est pas le cas, transmet uniquement la sortie du coordinateur de transactions externe à chaque gestionnaire de ressources participant.	Les opérations de lecture de validation en une phase et en deux phases sont admises. Les mises à jour en une phase ne sont pas admises sauf pour les transactions coordonnées par DB2 qui peuvent effectuer des mises à jour en une phase fédérées sur une connexion entrante en deux phases DRDA (Distributed Relational Database Architecture).	Éléments admis et gérés en fonction de règles de source de données de validation en une phase. Chaque instruction émise doit être la seule mise à jour dans une transaction de validation en une phase. Ne peut pas exister avec d'autres mises à jour de source de données de validation en deux phases dans la même transaction. Il est fortement recommandé que les instructions COMMIT ou ROLLBACK soient émises avant et après les transactions DDL transparentes.
En deux phases	Transaction globale DB2 de type 2 ou XA	Il peut s'agir du gestionnaire de transactions. Si ce n'est pas le cas, transmet uniquement la sortie du coordinateur de transactions externe à chaque gestionnaire de ressources participant.	Les opérations de lecture de validation en une phase et en deux phases sont admises. Plusieurs sources de données en deux phases peuvent être mises à jour.	Éléments admis et gérés en fonction des règles de source de données de validation en deux phases. Peut exister avec d'autres mises à jour de source de données de validation en une phase ou en deux phases dans la même transaction.

Conservation de l'atomicité et de la cohérences des données

Les serveurs fédérés tentent de garantir la cohérence des données et de conserver l'atomicité des transactions pour les sources de données.

Tout conflit entre un paramètre de point de synchronisation des applications et la fonction de mise à jour d'une source de données cible génère une erreur (SQL30090, code anomalie 18).

Les mises à jour locales incluant des éléments DDL de la base de données fédérée ne peuvent pas être associées dans la même transaction en tant que mise à jour d'une source de données en une phase fédérée. Transparent DDL

Utilisation d'instructions DDL et d'instructions DDL transparent

Les mises à jour locales incluant des éléments DDL de la base de données fédérée ne peuvent pas être associées dans la même transaction en tant que mise à jour d'une source de données en une phase fédérée. Les instructions DDL transparent constituent une exception. Pour les instructions DDL transparent, les mises à jour locales et les mises à jour de source de données sont admises, quel que soit le type de connexion et quelle que soit la validation (en une phase ou en deux phases).

Les instructions DDL transparentes créent une table sur une source de données éloignée et un pseudonyme dans la base de données fédérée pour la table éloignée. Un serveur fédéré traite les transactions DDL transparent comme des mises à jour.

Les instructions DDL transparent permettent de créer et de modifier des tables éloignées à l'aide du système de base de données DB2, sans qu'il soit nécessaire d'utiliser de session passe-système. Les instructions SQL pour le langage DDL transparent sont CREATE TABLE, ALTER TABLE et DROP TABLE. Par exemple, une instruction DDL transparent CREATE TABLE crée une table éloignée sur la source de données et un pseudonyme pour cette table sur le serveur fédéré. L'instruction contient une opération de mise à jour locale et une opération de mise à jour éloignée.

Certaines sources de données, telles Oracle, n'autorisent pas le langage DDL sur une connexion à validation en deux phases fédérées.

Activation d'une validation en deux phases pour les transactions fédérées

Pour utiliser la validation en deux phases pour des sources de données spécifiques, vous devez activer les serveurs fédérés associés. Le processus d'activation implique la préparation du serveur fédéré et la modification de la définition du serveur de source de données.

Avant de commencer

- Lorsque vous activez la validation en deux phases fédérées pour une source de données, vous augmentez le nombre d'enregistrements placés dans le journal de la base de données du serveur fédéré et dans le journal de la base de données de la source de données. Prenez en compte les conséquences sur l'administration et la maintenance de ces fichiers journaux afin de vous assurer que vos règles locales sont respectées.
- La source de données à utiliser doit être une source de données de validation en deux phases fédérées.

- La validation fédérée en deux phases n'est pas prise en charge dans l'environnement de traitement à parallélisme massif (MPP) et l'environnement isolé, à l'exception de Sybase. Pour Sybase sur UNIX uniquement, cette validation est prise en charge dans l'environnement isolé.
- Pour les sources de données DB2 for System i version 5.3 et antérieure et DB2 for z/OS, vérifiez que le paramètre de configuration SPM_NAME est défini avec la valeur par défaut, à savoir le nom d'hôte du serveur. SPM_NAME est par défaut une variante des sept premiers caractères du nom d'hôte TCP/IP. Avec DB2 for System i version 5.4 et ultérieure, vous n'avez pas besoin de définir SPM_NAME.

A propos de cette tâche

L'option de serveur DB2_TWO_PHASE_COMMIT active la validation en deux phases pour les sources de données. Vous enregistrez une définition de serveur de source de données à l'aide de l'instruction CREATE SERVER. La valeur définie pour DB2_TWO_PHASE_COMMIT est conservée pour toutes les connexions établies sous cette définition de serveur. Vous pouvez changer la valeur à tout moment à l'aide de l'instruction ALTER SERVER. Une fois l'instruction CREATE SERVER ou ALTER SERVER validée, le nouveau paramètre est disponible pour utilisation sur les requêtes de connexion sortantes suivantes.

Les clients et les applications client peuvent utiliser l'option SET SERVER OPTION pour remplacer temporairement la valeur en cours de l'option de serveur DB2_TWO_PHASE_COMMIT. L'instruction SET SERVER OPTION doit être exécutée immédiatement après la connexion à la base de données du serveur fédéré et avant que des connexions ne soient établies aux sources de données éloignées. La commande est effective pendant la durée de la connexion à la base de données fédérée. Vous ne pouvez pas changer l'option de serveur DB2_TWO_PHASE_COMMIT une fois que le serveur fédéré a établi une connexion à la source de données éloignée.

Lorsque vous incluez l'option XA_OPEN_STRING_OPTIONS dans une instruction CREATE SERVER, vous pouvez intégrer des informations spécialisées dans la chaîne XA_OPEN par défaut. Il peut s'agir de tout type d'information :

- ID uniques des transactions en plus des éléments fournis par IBM InfoSphere Federation Server
- Paramètres définis par l'utilisateur sur le mode de gestion des transactions
- Chaîne définie par l'utilisateur à ajouter à la requête XA_OPEN

Lorsqu'un appel XA_OPEN est effectué, généralement au début de la première transaction sur une source de données éloignée qui utilise la validation en deux phases, l'encapsuleur ajoute la valeur de la chaîne définie par l'utilisateur à la chaîne XA_OPEN par défaut pour l'appel XA_OPEN.

Vous pouvez inclure les options DB2_TWO_PHASE_COMMIT et XA_OPEN_STRING_OPTIONS dans une instruction CREATE SERVER, SET SERVER ou ALTER SERVER.

Procédure

Généralement, pour activer la validation en deux phases pour une source de données fédérée, procédez comme suit :

1. Exécutez l'instruction CREATE SERVER, ALTER SERVER ou SET SERVER avec l'option DB2_TWO_PHASE_COMMIT définie à Y.

2. Facultatif : Exécutez l'instruction CREATE SERVER, ALTER SERVER ou SET SERVER avec l'option XA_OPEN_STRING_OPTIONS.

Exemples d'option serveur

Cet exemple indique comment définir la validation en deux phases à l'aide de l'instruction CREATE SERVER :

```
CREATE SERVER Net8_Server TYPE ORACLE VERSION 8.1.7 WRAPPER NET8
OPTIONS (DB2_TWO_PHASE_COMMIT 'Y');
```

Cet exemple indique comment désactiver la validation en deux phases à l'aide de l'instruction ALTER SERVER :

```
ALTER SERVER Net8_Server OPTIONS (SET DB2_TWO_PHASE_COMMIT 'N');
```

Cet exemple indique comment définir un fichier de trace XA dans D:\Temp\sybase_xa.log pour l'encapsuleur Sybase à l'aide de l'option de serveur XA_OPEN_STRING_OPTIONS :

```
ALTER SERVER Ct1ib_Server OPTIONS (ADD XA_OPEN_STRING_OPTIONS
'-LD:\Temp\sybase_xa.log');
```

Cet exemple indique comment désactiver de manière temporaire la validation en deux phases à l'aide de l'instruction SET SERVER OPTION :

```
SET SERVER OPTION DB2_TWO_PHASE_COMMIT TO 'N' FOR SERVER Net8_Server;
```

Configuration requise pour la source de données et pour les transactions de validation en deux phases fédérées

Avant d'activer la validation en deux phases fédérées pour une source de données, vous devez vous assurer qu'il s'agit d'une source de données prise en charge.

Les systèmes fédérés prennent en charge les opérations de validation en deux phases avec les sources de données suivantes :

- Sources de données de la famille DB2 via le protocole DRDA (Distributed Relational Database Architecture) :
 - DB2 Universal Database for Linux, UNIX et Windows, version 8.1 ou ultérieure
 - DB2 Universal Database for z/OS, version 7.1 ou ultérieure
 - DB2 Universal Database for System i version 5.3 ou ultérieure
- Informix IDS, version 7.31 ou version ultérieure, version 9.40 ou ultérieure, version 10.0 ou ultérieure
- Informix XPS, version 8.40 ou ultérieure
- Microsoft SQL Server 2000 et Microsoft SQL Server 2005 pour un serveur fédéré uniquement sur Windows
- Oracle, version 8.1.7 ou ultérieure avec la bibliothèque XA
- Sybase Adaptive Server Enterprise, version 12 ou ultérieure, avec la bibliothèque XA pour un serveur fédéré uniquement sur Windows

Si vous tentez d'activer la validation en deux phases fédérées pour une source de données non prise en charge, vous recevez une erreur SQL1881N.

Configuration des sources de données DRDA

Le serveur fédéré offre la connectivité aux sources de données DB2 en utilisant le protocole DRDA ouvert. Cette prise en charge est équivalente à celle offerte par le serveur DB2 Connect.

De plus, pour la validation en deux phases, le serveur fédéré interagit avec chaque source de données à l'aide du modèle XA.

Avant de commencer

Restrictions :

- Toutes les sources de données DB2 ne prennent pas en charge XA sur DRDA de manière native. Pour ces dernières, comme DB2 for z/OS et DB2 for System i, le serveur fédéré utilise le gestionnaire de point synchronisé (SPM). Ce dernier effectue un mappage entre les flux de validation en deux phases XA et non XA pris en charge par tous serveurs DB2. Lorsque la validation en deux phases fédérées est fournie via le gestionnaire de points de synchronisation, toutes les sémantiques XA ne sont pas prises en charge car il existe des incompatibilités entre la prise en charge fédérée et le gestionnaire de points de synchronisation. Par exemple, les transactions ne peuvent pas être imbriquées. Toutes les transactions doivent être validées ou annulées avant le démarrage d'une nouvelle transaction.
- La validation en deux phases fédérées prend en charge DB2 for z/OS, mais cette application n'accepte pas l'exécution d'une instruction SAVEPOINT dans une transaction de validation en deux phases fédérée.
- Dans une transaction gérée par DB2, un client DB2 for z/OS peut réaliser une mise à jour en une phase fédérée sur une connexion entrante en deux phases DRDA. Toutefois, une telle mise à jour ne peut pas être effectuée via une connexion entrante en deux phases XA DRDA. Il n'est pas possible d'effectuer des mises à jour en une phase et en deux phases via une connexion entrante en deux phases DRDA.
- L'option de serveur XA_OPEN_STRING_OPTIONS n'est pas prise en charge pour les sources de données DRDA. Si vous utilisez l'option, une erreur SQL1881 est renvoyée.

Configuration requise :

- Pour les sources de données DB2 qui prennent en charge XA en utilisant SPM en non en prenant en charge XA de manière native, vérifiez que les valeurs par défaut sont attribuées aux paramètres SPM_NAME et SVCENAME de la configuration du gestionnaire de bases de données.

Procédure

Pour configurer une source de données DRDA :

Exécutez l'instruction CREATE SERVER, ALTER SERVER ou SET SERVER avec l'option DB2_TWO_PHASE_COMMIT définie à Y.

L'encapsuleur DRDA génère automatiquement la chaîne XA OPEN suivante pour les sources de données DRDA :

```
DB=nombasededonnées,UID=idutilisateur,PWD=motdepasse,TPM=FDB2,HOLD_CURSOR=T
```

Configuration des sources de données Oracle

Il existe plusieurs exigences et restrictions pour l'utilisation des sources de données Oracle en vue de la validation en deux phases fédérées.

Avant de commencer

Restrictions :

- DDL passe-système et DDL transparent dirigés vers Oracle n'aboutissent pas avec le code anomalie SQL30090 21 (ORA-2089) SQL régulier soumis dans des sessions passe-système fonctionne.

Configuration requise :

- Le client Oracle utilisé sur le serveur fédéré doit être une installation complète afin de garantir que toutes les bibliothèques liées à XA sont présentes. Vérifiez que l'exécution de la commande `djxlinkOracle` a abouti afin que toutes les bibliothèques DB2 et Oracle soient disponibles et correctement liées. Prenez en compte le fait que les scripts `djxlink` sont exécutés automatiquement si vous installez le client Oracle avant d'installer le serveur fédéré.
- Vous devez accorder les privilèges suivants à tous les utilisateurs qui exécutent des transactions de validation en deux phases à partir du serveur fédéré :
 - `grant select on dba_pending_transactions to USERID;`
 - `grant select on dba_2pc_pending to USERID;`
 - `grant force transaction to USERID;`
- Vous pouvez également accorder les privilèges suivants aux utilisateurs qui exécutent des transactions de validation en deux phases à partir du serveur fédéré :
 - `grant force any transaction to USERID;`
- Si vous avez l'intention d'exécuter plus de dix transactions de validation en deux phases de manière simultanée, pensez à augmenter le paramètre `distributed_transactions` du serveur Oracle se trouvant dans le fichier `init.ora`.

Procédure

Pour configurer une source de données Oracle :

1. Exécutez l'instruction `CREATE SERVER`, `ALTER SERVER` ou `SET SERVER` avec l'option `DB2_TWO_PHASE_COMMIT` définie à Y.

L'encapsuleur Oracle crée automatiquement la chaîne XA OPEN suivante pour les sources de données Oracle :

```
Oracle_XA=Acc=Puid/password+Seq=0+DB=dbname+SqlNet=dblink+Threads=true
```

Par exemple :

```
XA_OPEN_STRING_OPTIONS '+LogDir=/home/user/directory+DbgFl=0x7'
```

2. Indiquez des options XA supplémentaires à l'aide de l'option de serveur `XA_OPEN_STRING_OPTIONS`.

Configuration des sources de données Informix

Il existe plusieurs exigences et restrictions pour l'utilisation des sources de données Informix en vue de la validation en deux phases fédérées.

Avant de commencer

Restrictions :

- Vous ne pouvez pas accéder aux pseudonymes Informix avec un serveur de validation en deux phases et un serveur de validation en une phase dans une seule connexion au serveur fédéré.
- L'option de curseur `WITH HOLD` n'est pas prise en charge.

- L'option de serveur XA_OPEN_STRING_OPTIONS n'est pas prise en charge pour les sources de données Informix.

Configuration requise :

- La journalisation doit être activée pour la base de données Informix.
- La bibliothèque Informix XA permet uniquement une connexion par unité d'exécution. Par conséquent, le serveur fédéré ne peut pas accéder aux sources de données Informix avec plusieurs serveurs qui sont activées pour la validation en deux phases fédérées dans une seule connexion. Si une application doit utiliser plusieurs serveurs qui sont activés pour la validation en deux phases fédérées, effectuez les étapes facultatives dans la procédure suivante.

Procédure

Pour configurer une source de données Informix :

1. Exécutez l'instruction CREATE SERVER, ALTER SERVER ou SET SERVER avec l'option DB2_TWO_PHASE_COMMIT définie à Y.

L'encapsuleur Informix génère la chaîne XA OPEN suivante pour les sources de données Informix :

```
DB=nombasededonnées;RM=nomrm;CON=con;USER=utilisateur;PASSWD=motdepasse
```

2. Si une application doit utiliser plusieurs serveurs qui sont activées pour la validation en deux phases fédérées, suivez la procédure ci-après.

- a. Copiez les bibliothèques de l'encapsuleur Informix : libdb2informix.a, libdb2informixF.a et libdb2informixU.a.
- b. Définissez plusieurs instances de l'encapsuleur Informix en indiquant une copie différente des bibliothèques de l'encapsuleur Informix dans la clause LIBRARY de l'instruction CREATE SERVER.
- c. Définissez chaque serveur de validation en deux phases fédérées pour les différentes instances d'encapsuleur.

Par exemple :

```
CREATE WRAPPER wrapper1 library 'libdb2informix.a'
CREATE SERVER server1 type informix version 9.4 wrapper wrapper1 options
(node 'inf1', dbname 'firstdb', db2_two_phase_commit 'Y');
CREATE WRAPPER wrapper2 library 'libdb2informix2.a'
CREATE SERVER server2 type informix version 9.4 wrapper wrapper2 options
(node 'inf2', dbname 'seconddb', db2_two_phase_commit 'Y');
```

Configuration des sources de données Microsoft SQL Server

Il existe plusieurs exigences et restrictions pour l'utilisation des sources de données Microsoft SQL Server en vue de la validation en deux phases fédérées.

Avant de commencer

Restrictions :

- Pour les sources de données Microsoft SQL Server, la validation en deux phases fédérées est prise en charge uniquement par IBM InfoSphere Federation Server installé sous Windows.
- Le niveau d'isolement DB2 n'est pas propagé sur le serveur Microsoft SQL.

Configuration requise :

- Pour que la validation en deux phases fédérées fonctionne avec Microsoft SQL, une option de serveur supplémentaire, XA_OPEN_STRING_OPTIONS, doit être ajoutée au serveur :


```
alter server S1 options(add xa_open_string_options
'RMRecoveryGuid=c200e360-38c5-11ce-ae62-08002b2b79ef');
```

où RMRecoveryGuid = ID gestionnaire de ressources.

L'ID gestionnaire est disponible à l'emplacement suivant du registre Microsoft SQL Server :

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer]
"ResourceMgrID" = "{resource manager ID}"
```

Procédure

Pour configurer une source de données Microsoft SQL Server :

1. Exécutez l'instruction CREATE SERVER, ALTER SERVER ou SET SERVER avec l'option DB2_TWO_PHASE_COMMIT définie à Y.

L'encapsuleur Microsoft SQL Server génère automatiquement la chaîne XA OPEN suivante pour les sources de données Microsoft SQL Server :

```
TM=nomtm
```

2. Indiquez des options XA supplémentaires en plus de la valeur RMRecovery requise à l'aide de l'option de serveur XA_OPEN_STRING_OPTIONS.

Configuration des sources de données Sybase

Il existe plusieurs exigences et restrictions pour l'utilisation des sources de données Sybase en vue de la validation en deux phases fédérées.

Avant de commencer

Restrictions :

- Pour les sources de données Sybase, la validation en deux phases fédérées est prise en charge uniquement par IBM InfoSphere Federation Server installé sous Windows.
- DDL passe-système et DDL transparent dirigés vers Sybase n'aboutissent pas avec une erreur SQL910N. SQL régulier soumis dans des sessions passe-système fonctionne.

Configuration requise :

- L'administrateur de bases de données Sybase doit avoir une licence pour la gestion des transactions réparties de Sybase ASE (Adaptive Server Enterprise) et doit activer la fonction avec la commande suivante dans l'outil isql :

```
sp_configure 'enable dtm', 1
```

Sybase ASE doit être redémarré pour que ce paramètre soit pris en compte.

- Le nom d'utilisateur indiqué dans une chaîne ouverte doit avoir l'élément dtm_tm_role dans Sybase ASE correspondant. L'administrateur peut attribuer ce rôle à l'aide de la commande suivante dans l'outil isql :

```
sp_role "grant", dtm_tm_role, user_name
```

- Pour qu'une source de données Sybase ASE (Adaptive Server Enterprise) se comporte comme un gestionnaire de ressources sur une base de données fédérée, une entrée du gestionnaire de ressources logiques (LRM) doit exister dans le fichier xa_config du répertoire \$SYBASE/\$SYBASE_OCS/config (version 12 ou ultérieure) qui mappe le nom du gestionnaire de ressources vers le nom Sybase ASE. Pour plus d'informations, consultez la documentation Sybase ASE XA.

Le nom LRM est utilisé par le serveur fédéré dans la chaîne XA OPEN. Le serveur fédéré utilise le nom du noeud Sybase ASE pour le nom LRM.

- Vérifiez que le nom de serveur indiqué dans le fichier de configuration XA xa_config se trouve dans le fichier d'initialisation sql.ini du répertoire \$SYBASE/ini.

Procédure

Pour configurer une source de données Sybase :

1. Installez le fichier de bibliothèque Sybase XA libxadtm.dll sur le serveur fédéré.
2. Avant d'utiliser les fonctions de validation en deux phases fédérées, créez les entrées LRM suivantes dans le fichier \$SYBASE/\$SYBASE_OCS/config/xa_config. Si vous n'avez de droits en écriture pour le fichier xa_config, créez un fichier xa_config dans un autre répertoire et définissez son chemin absolu dans la variable d'environnement XACONFIGFILE du fichier db2dj.ini :

```
;une ligne
de commentaire est requise
lrm=nom_lrm
server=nom_serveur
```

où *nom_serveur* correspond au nom d'entrée dans le fichier \$SYBASE/ini/sql.ini.

3. L'encapsuleur Sybase crée automatiquement la chaîne XA_OPEN par défaut suivante pour les sources de données Sybase :

```
-Nnomrm -Uidutilisateur -Pmotdepasse
```

Si vous devez indiquer d'autres options pour la chaîne XA_OPEN, utilisez l'option de serveur XA_OPEN_STRING_OPTIONS.

Reprise suite à des incidents de validation en deux phases fédérées

Un système fédéré peut récupérer suite à des incidents lors de la validation en deux phases avec la resynchronisation automatique ou la reprise manuelle des transactions en attente de validation.

Resynchronisation pour les systèmes fédérés

La validation en deux phases fédérées inclut un processus automatique qui tente de gérer les erreurs lors des transactions de validation.

Outre les erreurs sur le serveur fédéré, un environnement fédéré augmente le risque d'erreurs pouvant survenir suite à des erreurs de source de données, de communications ou réseau.

Pour garantir l'intégrité des données, le serveur fédéré gère ces erreurs lors du processus de validation en deux phases fédérées :

Erreur de la première phase

Si une base de données indique qu'elle n'a pas pu préparer la validation d'une unité d'oeuvre, le serveur fédéré annule l'unité d'oeuvre lors de la deuxième phase du processus de validation. Lors de la deuxième phase, le serveur fédéré envoie un message d'annulation à toutes les sources de données participantes attendant la sortie des transactions.

Erreur de la deuxième phase

La gestion des erreurs à cette phase varie selon que la deuxième phase valide ou annule la transaction. La deuxième phase annule uniquement la transaction si une erreur est survenue dans la première phase.

Si une des sources de données participantes ne peut pas valider ou annuler l'unité d'oeuvre, suite à une erreur de communication, le serveur fédéré tente à nouveau la validation ou l'annulation via un processus appelé resynchronisation. La resynchronisation est lancée et gérée automatiquement par le serveur fédéré. L'application à l'origine des appels est informée que la validation a abouti via la zone SQLCA (SQL communications area) si l'application se connecte au serveur fédéré via une connexion de validation en deux phases. L'application à l'origine des appels est déconnectée du serveur fédéré si l'application se connecte au serveur fédéré via une connexion de validation en une phase.

La plupart des sources de données ne peuvent pas lancer la resynchronisation si une erreur s'est produite dans un système fédéré. Le serveur fédéré initie le processus de resynchronisation.

Dans certains cas, un serveur fédéré peut être en situation d'échec lors du traitement des transactions, suite à une coupure d'alimentation, par exemple. Généralement, la resynchronisation résout les transactions d'unité d'oeuvre sans intervention.

La resynchronisation tente de mener à terme toutes les transactions en attente de validation. Lors de la resynchronisation normale, l'agent de resynchronisation se connecte à la base de données du gestionnaire de transactions et émet une décision de validation ou d'annulation. Le gestionnaire de transactions fédérées propage alors cette décision aux sources de données ayant pris part à la transaction d'unité d'oeuvre répartie.

Reprise manuelle de transactions en attente de validation

Si vous ne pouvez pas attendre que la resynchronisation résolve automatiquement les transactions en attente de validation, vous pouvez effectuer cette opération manuellement. Ce processus est parfois appelé traitement heuristique.

Par exemple, le lien de communications entre le gestionnaire de ressources externes et un gestionnaire de transactions fédérées n'aboutit pas lors d'une transaction. Si vous disposez de suffisamment d'informations sur la transaction, vous pouvez libérer des ressources sur le serveur fédéré et sur les sources de données en annulant la transaction du serveur fédéré.

Utilisez le traitement heuristique uniquement lorsque vous connaissez la raison de l'erreur de la transaction. Vous devez également libérer immédiatement les ressources verrouillées. Dans la plupart des cas, laissez la resynchronisation automatisée récupérer les transactions. Il existe plusieurs couches de gestion des transactions dans un système fédéré. La récupération des transactions de manière heuristique est un processus complexe et pouvant entraîner des risques.

Il existe plusieurs moyens de base d'effectuer le traitement heuristique :

- Commande LIST INDOUBT TRANSACTIONS
Vous pouvez utiliser cette commande pour le traitement heuristique.
- Fenêtre Gestionnaire de transactions en attente de validation
Vous pouvez utiliser cet outil d'interface utilisateur graphique pour le traitement heuristique.
- API heuristiques
Vous pouvez utiliser ces API dans vos applications pour effectuer le traitement heuristique.

Les tâches et les opérations spécifiques que vous utilisez pour effectuer le traitement heuristique varient en fonction des circonstances de l'erreur.

Dans des systèmes fédérés, lorsqu'une demande de traitement heuristique est envoyée à un gestionnaire de transactions fédérées, la décision en résultant de validation ou d'annulation doit être compatible avec le statut réel des transactions en attente de validation sur le serveur fédéré. Sinon, un message d'erreur est renvoyé.

Le statut de la validation en deux phases fédérées est légèrement différente de la validation en deux phases DB2 des transactions en attente de validation :

- Le statut (d) indique qu'un accusé de réception de validation de la transaction manque provenant d'une ou plusieurs sources de données fédérées.
- Le statut (b) indique qu'un accusé de réception d'annulation de la transaction manque provenant d'une ou de plusieurs sources de données fédérées.

Si vous ne pouvez pas valider ou annuler une transaction avec le statut (d) ou (b), vous pouvez indiquer que la transaction doit être ignorée en utilisant l'option (f). Toutefois, lorsque vous utilisez l'option (f), tous les enregistrements de la transaction sont retirés du serveur fédéré et vous devez manuellement nettoyer les incidents de synchronisation restants sur les sources de données impliquées. Utilisez uniquement l'option (f) lorsqu'elle est absolument nécessaire, par exemple en cas de panne d'un serveur éloigné ou lorsque des connexions aux serveurs éloignés sont supprimées, et lorsqu'il est urgent de libérer des ressources. Utilisez cette option avec vigilance.

Remarque : Etant donné que les statuts (d) et (b) sont nouveaux pour WebSphere Federation Server v9.1, les clients fédérés de niveau inférieur ne peuvent pas les prendre en charge. Si vous utilisez un client de niveau inférieur pour récupérer manuellement des transactions en attente de validation, les statuts (d) et (b) sont mappés vers le statut (m), à la place, qui n'est pas une valeur appropriée. Pour empêcher des informations inappropriées lorsque vous récupérez manuellement des transactions en attente de validation, vérifiez que vous utilisez un client fédéré v9.1. Par défaut, l'ordinateur qui exécute WebSphere Federated Server v9.1 inclut le client fédéré v9.1.

Traçage des états des transactions d'unité d'oeuvre répartie dans des sources de données

Si vous décidez de résoudre manuellement des transactions en attente de validation au lieu de laisser la synchronisation les résoudre automatiquement, il est essentiel d'effectuer le traçage des transactions dans le système fédéré. Lorsque vous tracez une transaction d'unité d'oeuvre répartie en attente de validation, la seule manière de déterminer la ou les sources de données n'ayant pas abouti consiste à capturer l'élément XID pour la transaction en situation d'échec.

Vous devez rechercher cet XID dans les gestionnaires de sources de données pour l'ensemble des sources de données auxquelles un serveur fédéré peut avoir accédé en tant que partie de la transaction d'unité d'oeuvre répartie.

Pour déterminer l'identifiant et l'état de chaque transaction impliquée dans l'unité d'oeuvre répartie à tracer, émettez la commande LIST INDOUBT TRANSACTIONS dans la base de données d'applications, dans la base de données fédérée et dans toute source de données de la transaction de l'unité d'oeuvre répartie.

Remarque : Chaque source de données qui prend en charge le protocole de validation en deux phases peut utiliser une commande différente. Recherchez la chaîne XID dans la documentation relative aux commandes de la source de données spécifique.

Le gestionnaire de transactions fédérées génère un XID au format hexadécimal lors du traitement des transactions. Cet XID commence par l'identificateur de format F2PC, qui est le chiffre 46325243 au format hexadécimal. Le gestionnaire de transactions fédérées envoie le XID aux sources de données. Toutefois, avant cela, le serveur fédéré modifie cet élément afin qu'il soit en accord avec le format XID de cette source de données. Ces modifications incluent la mise à jour de la section de qualificatif de branche du XID et l'ajout de cette section au XID.

Il peut être nécessaire de comparer les différents XID dans les sources de données, vous devez donc connaître quelle est la partie du XID à comparer dans votre environnement lorsque vous tracez un XID.

Par exemple, le gestionnaire de transactions est une base de données DB2. Lorsque vous émettez la commande LIST INDOUBT sur la base de données, une chaîne en représentation hexadécimale est renvoyé pour le XID transaction similaire à l'élément suivant :

```
4632504300000019 000000004739314533463135 2E47453934000000 000000000000E80000
```

Cette chaîne est composée de plusieurs parties :

ID format	Longueur de l'identifiant de transaction	Longueur de l'identifiant de branche	Identifiant de transaction
46325043	00000019	00000000	4739314533463135 2E47453934000000 000000000000E800 00

Les valeurs répertoriées dans cette chaîne sont hexadécimales. Par exemple, la longueur de l'identifiant de transaction (hexadécimal 19) représente la valeur décimale 25.

Si le même XID est transmis d'un serveur fédéré qui se comporte comme un gestionnaire de transactions fédérées à une source de données qui se comporte comme un gestionnaire de ressources, la chaîne XID est ajoutée. Par exemple, l'élément XID est transmis à un gestionnaire de ressources pour un système de base de données DB2 pour une source de données Windows au format suivant :

```
4632524300000019 000000014739314533463135 2E47453934000000 000000000000E8000001
```

Voici la même chaîne XID séparée en plusieurs parties :

ID format	Longueur TID	Longueur de l'identifiant de branche	Identifiant de transaction	Qualificatif de branche
46325043	00000019	00000001	4739314533463135 2E47453934000000 000000000000E800 00	01

La longueur de qualificatif de branche est 1 et la section de qualificatif de branche est ajoutée. Toutefois, la zone d'identifiant n'a pas changée. Vous pouvez toujours tracer l'élément XID sur différentes sources de données en limitant la chaîne de recherche à la section de l'identifiant de transaction.

Résolution des incidents de validation en deux phases fédérées

La résolution des incidents de validation en deux phases fédérées est souvent propre à la source de données dans laquelle survient l'incident.

Les applications doivent gérer les codes d'erreur qui indiquent que les transactions ont expiré, plus particulièrement les codes d'erreur -913 et -918. Elles doivent également vérifier si des codes d'erreur -911 sont générés.

Résolution des incidents de la source de données Oracle

Essayez les méthodes suivantes pour résoudre les incidents survenant dans les sources de données Oracle.

- Pour journaliser les informations :

```
db2 "alter server oral options (add XA_OPEN_STRING_OPTIONS  
'+LogDir=C:\temp+DbgFl=0x7')
```

où C:\temp correspond au chemin complet du fichier journal créé. La journalisation des informations peut ralentir les performances de manière importante, journalisez des informations uniquement lorsque vous résolvez les incidents.

- Pour afficher les informations de trace, ajoutez les lignes suivantes au fichier sqlnet.ora client Oracle :

```
TRACE_LEVEL_CLIENT=16  
TRACE_DIRECTORY_CLIENT=C:\temp
```

Vous pouvez également attribuer un nombre inférieur à TRACE_LEVEL_CLIENT, tel que 4 ou 8. L'affichage des informations de trace peut ralentir les performances de manière importante, activez les informations de niveau de trace uniquement lorsque vous résolvez les incidents.

- Pour répertorier les transactions en attente sur la source de données Oracle :

```
select * from dba_pending_transactions where formatid=1177702467;  
select * from dba_2pc_pending;
```

Pour répertorier les états d'une transaction et les ID d'une transaction :

```
select A.STATE, A.LOCAL_TRAN_ID, A.FAIL_TIME, A.GLOBAL_TRAN_ID,  
B.FORMATID || '.' || B.GLOBALID || '.' || b.BRANCHID as fmt_xid  
from dba_2pc_pending A, dba_pending_transactions B  
where A.GLOBAL_TRAN_ID = B.FORMATID || '.' ||  
B.GLOBALID and STATE='prepared' and B.FORMATID=1177702467;
```

- Pour résoudre manuellement les transactions :

```
rollback force '4.31.157818';  
commit force '10.24.154537'
```

où '4.31.157818' est la zone Oracle A.LOCAL_TRAN_ID qui correspond à l'élément XID dans GLOBAL_TRAN_ID.

Résolution des incidents de la source de données Sybase

Essayez les méthodes suivantes pour résoudre les incidents survenant dans les sources de données Sybase.

- Consultez le fichier journal Sybase XA `syb_xa_log` se trouvant dans le répertoire `$SYBASE`.
- Utilisez l'outil `db2diag` pour consulter le fichier `db2diag.log`.
- Pour consulter une transaction sur le serveur Sybase :

```
$ isql -Uuser_name -Ppassword -Sserver_name  
1> transactions_sp  
2> go
```

Si vous trouvez une transaction non valide ou superflue, demandez à l'administrateur Sybase de supprimer la transaction.

Performances de la validation en deux phases fédérées

Les sources de données qui sont configurées pour les transactions de validation en deux phases provoquent une baisse des performances lorsque vous les comparez aux sources de données configurées pour des transactions de validation en une phase.

Lorsqu'une source de données utilise une validation en deux phases, le serveur fédéré se comporte comme un coordinateur, garantissant que tous les participants sont correctement synchronisés. Cette coordination est terminée suite à la journalisation supplémentaire sur le serveur fédéré et des communications supplémentaires entre les sources de données. Par conséquent, le traitement requis pour une transaction fédérée accédant à une source de données pour la validation en deux phases est plus important que pour une transaction qui accède à une source de données pour une validation en une phase. Une source de données doit ensuite être activée pour la validation en deux phases uniquement lorsque les transactions fédérées requièrent une validation en deux phases.

Une transaction fédérée qui requiert une validation en une phase, telle une mise à jour sur un seul site, mais qui s'exécute sur une source de données pour la validation en deux phases peut subir une baisse des performances lorsque vous la comparez à la même transaction exécutée sur une source de données sans la validation en deux phases.

Une transaction sous le contrôle de validation en deux phases subit le traitement supplémentaire suivant, que la transaction requiert ou non une validation en deux phases :

1. Toutes les transactions requièrent qu'un journal de base de données supplémentaire soit placé dans le fichier journal du serveur.
Les écritures de journal supplémentaires permettent au serveur fédéré d'effectuer le suivi des sources de données dans la transaction afin de coordonner les opérations de validation et d'annulation suivantes.
2. Toutes les transactions requièrent des communications de base de données supplémentaires entre le serveur fédéré et la source de données.
3. Les opérations d'insertion, de mise à jour et de suppression requièrent une ou plusieurs écritures de journal de base de données supplémentaires sur la source de données éloignée.

La plus grande partie du traitement supplémentaire se produit dans les limites de la transaction et n'est pas influencée par le contenu de la transaction. Par conséquent, l'augmentation du pourcentage de temps écoulé pendant une courte transaction est supérieure à celle d'une longue transaction.

Les transactions simultanées font que le serveur fédéré place plusieurs enregistrements de journal à partir de la mémoire tampon des journaux dans le fichier journal en une seule opération d'écriture. Par conséquent, les applications exécutant les transactions de validation en deux phases subissent de manière simultanée un temps système inférieur aux applications exécutant les mêmes transactions en série.

Amélioration des performances de validation en deux phases fédérées

Vous pouvez améliorer les performances des transactions dans une configuration de validation en deux phases fédérées.

Prenez en compte les configurations possibles suivantes :

- Afin de réduire le temps passé à écrire des enregistrements de journal supplémentaires dans le serveur fédéré, placez les fichiers journaux du serveur fédéré sur une unité gérant des transactions d'écriture rapides, de préférence une unité ayant une mémoire cache d'écriture. Généralement, les transactions d'écriture de journal du serveur fédéré sont à l'origine du temps de traitement supplémentaire. La correction du placement des fichiers journaux peut améliorer les performances de la validation en deux phases. Cette amélioration est particulièrement notable pour les transactions en lecture seule. Le placement des fichiers journaux du serveur fédéré est contrôlé par le paramètre de configuration de la base de données NEWLOGPATH.
- Pour les transactions d'insertion, de mise à jour et de suppression, placez les fichiers journaux de la source de données sur un support acceptant les écritures rapides.
- Pour réduire le temps système généré par des messages XA supplémentaires envoyés entre le serveur fédéré et les sources de données :
 - Placez le serveur fédéré sur le même ordinateur que les sources de données de validation en deux phases.
 - Si vous ne pouvez pas placer le serveur fédéré sur le même ordinateur que la source de données, augmentez la vitesse du réseau et réduisez l'attente entre le serveur fédéré et les sources de données afin d'améliorer les performances.

Pour les applications, activez uniquement la validation en deux phases pour une source de données lorsqu'au moins une transaction de l'application requiert la validation en deux phases. Pour chaque serveur, attribuez la valeur N à la valeur par défaut de DB2_TWO_PHASE_COMMIT et utilisez l'instruction SET SERVER OPTION pour activer la validation en deux phases dans les applications nécessitant spécifiquement ce type de validation.

Chapitre 10. Insertion, mise à jour et suppression des données dans un système fédéré

Lors du développement de votre environnement fédéré, selon que vous choisissez de modifier les informations distantes ou de déplacer les données vers des données source, vous devez insérer, mettre à jour ou supprimer les données sur vos sources de données.

Avant de modifier des données sur une source de données distante, assurez-vous de disposer des privilèges d'autorisation appropriés pour émettre les instructions INSERT, UPDATE et DELETE sur le pseudonyme. Vous devez disposer de connaissances en matière d'intégrité référentielle, de préservation de l'atomicité de l'instruction et de sémantique d'affectation.

Privilèges des autorisations pour les instructions INSERT, UPDATE et DELETE

Les privilèges requis pour l'émission d'instructions INSERT, UPDATE et DELETE sur des pseudonymes sont similaires aux privilèges requis pour l'émission de ces mêmes instructions sur des tables. De plus, vous devez conserver des privilèges appropriés sur la source de données afin d'effectuer des opérations de sélection, d'insertion, de mise à jour et de suppression sur l'objet sous-jacent.

Vous pouvez accorder ou révoquer des privilèges SELECT, INSERT, UPDATE et DELETE pour un pseudonyme.

Toutefois, l'octroi ou la révocation de privilèges pour un pseudonyme n'accorde pas ou ne révoque pas de privilèges sur la source de données. Sur la source de données, des privilèges doivent être accordés ou révoqués pour l'autorisation REMOTE_AUTHID indiquée dans le mappage utilisateur sur le serveur fédéré.

Les privilèges détenus par l'ID autorisation de l'instruction doivent inclure les privilèges requis pour le pseudonyme (pour que la base de données fédérée accepte la requête). L'ID utilisateur sur la source de données mappé vers l'ID utilisateur (via un mappage utilisateur) doit disposer des privilèges requis sur l'objet de table sous-jacent (pour que la source de données accepte la requête).

Lorsqu'une requête est soumise à la base de données fédérée, les privilèges des autorisations concernant le pseudonyme dans la requête sont vérifiés. Les autorisations requises pour l'objet de source de données référencé par le pseudonyme sont appliquées uniquement lors du traitement de la requête. Si aucun privilège SELECT n'est attribué au pseudonyme, vous ne pouvez pas effectuer de sélection dans l'objet de source de données auquel le pseudonyme fait référence.

De la même manière, si vous disposez d'un privilège UPDATE sur le pseudonyme, cela ne signifie pas vous serez automatiquement autorisé à mettre à jour l'objet de source de données que le pseudonyme représente. La transmission de la vérification des privilèges sur le serveur fédéré n'implique pas la transmission de la vérification des privilèges sur la source de données distante. Via les mappages

utilisateur, un ID autorisation de serveur fédéré est mappé vers l'ID utilisateur de la source de données. La vérification des privilèges est appliquée sur la source de données.

Restrictions INSERT, UPDATE et DELETE du système fédéré

Certaines restrictions s'appliquent à l'utilisation des instructions INSERT, UPDATE ou DELETE dans un système fédéré.

Les restrictions suivantes s'appliquent à la mise à jour des pseudonymes :

- Il n'est pas possible de mettre à jour un objet de source de données en lecture seule, tel une vue JOIN.
- Vous ne pouvez pas effectuer d'opérations d'insertion, de mise à jour et de suppression sur des vues fédérées créées à l'aide d'instructions UNION ALL. Les vues fédérées créées à l'aide d'instructions UNION sont des vues en lecture seule.
- Une opération d'insertion, de mise à jour ou de suppression fédérée, ou l'appel d'une procédure fédérée avec une indication d'accès aux données SQL de MODIFIES SQL DATA n'est pas valide dans une fonction, une option données-modification-table-référence, une instruction composée qui indique ATOMIC (à l'exception de la source de données DB2 for Linux, UNIX et Windows), un déclencheur et un environnement d'exécution d'applications où l'une conditions suivantes est vérifiée :
 - La commande SAVEPOINT est effective (à l'exception de la table DB2 pour la source de données Linux, UNIX et Windows)
 - Le curseur défilable est utilisé
 - La vue cible contient plusieurs tables ou pseudonymes

Sources de données non prises en charge

Le système fédéré ne prend pas en charge les opérations d'insertion, de mise à jour et de suppression pour les pseudonymes sur les sources de données non relationnelles.

Les sources de données non prises en charge par la fédération incluent :

- BioRS
- Excel
- Fichiers structurés sous forme de table
- services Web
- XML

Intégrité référentielle dans un système fédéré

Dans un système fédéré, la base de données fédérée n'applique pas l'intégrité référentielle dans les sources de données.

Toutefois, les contraintes d'intégrité référentielle sur une source de données peuvent affecter les mises à jour de pseudonymes. Supposons que vous deviez insérer des données d'un serveur fédéré dans un pseudonyme. Lorsque le serveur fédéré envoie l'insertion à la source de données, il ne respecte pas une contrainte d'intégrité référentielle sur la source de données. Le serveur fédéré mappe l'erreur générée vers une erreur fédérée.

Les applications sont chargées de l'intégrité référentielle entre les sources de données.

Instructions INSERT, UPDATE et DELETE et objets LOB

Vous pouvez effectuer des opérations de lecture pour des objets LOB sur des systèmes fédérés. Les opérations d'écriture sur des objets LOB sont prises en charge sur certaines sources de données.

A l'aide de la fédération, vous pouvez effectuer des opérations de lecture sur des objets LOB se trouvant dans une source de données relationnelles. Vous pouvez effectuer des opérations d'écriture sur des objets LOB se trouvant dans les sources de données suivantes :

- Oracle, avec l'encapsuleur NET8
- DB2 for z/OS, DB2 for System i DB2 Database for Linux, UNIX et Windows avec l'encapsuleur DRDA

Sous certaines conditions, vous pouvez effectuer des opérations d'écriture sur des objets LOB dans d'autres sources de données en modifiant le type de colonne de pseudonyme en élément VARCHAR.

Conservation de l'atomicité des instructions dans un système fédéré

Lors des opérations de mise à jour, les systèmes fédérés tentent toujours de conserver des données à l'état atomique lors de l'exécution d'une instruction DML. Lorsque les données sont à l'état atomique, il est garanti que leur traitement aboutisse ou qu'elles restent inchangées.

Lorsqu'un client ou une application émet une instruction INSERT, UPDATE ou DELETE sur un pseudonyme, un serveur de base de données fédéré traite en interne cette instruction en tant qu'instruction DML unique ou en tant qu'ensemble d'instructions DML. Si un serveur fédéré doit envoyer plusieurs instructions DML à une source de données cible pour traitement, il est possible que l'atomicité des données soit compromise. Pour empêcher la compromission de l'atomicité des données, les systèmes fédérés utilisent les API du point de sauvegarde de la source de données afin de contrôler l'ensemble des instructions DML.

Certaines sources de données n'externalisent pas les API de point de sauvegarde que le système fédéré peut utiliser. Dans ces conditions, les instructions INSERT, UPDATE ou DELETE sont exécutées sans la protection des API de point de sauvegarde.

Lorsqu'une erreur se produit dans des transactions d'insertion, de mise à jour et de suppression fédérées, les résultats de mise à jour partiels peuvent être générés sur la source de données. Pour corriger les incidents d'incohérence, un système fédéré effectue automatiquement une annulation de transaction interne avant de renvoyer une erreur SQLCODE aux applications.

Les sources de données suivantes n'externalisent pas les API de point de sauvegarde que le serveur fédéré peut utiliser :

- DB2 for System i
- DB2 for VM et VSE
- Informix
- JDBC
- Microsoft SQL Server

- ODBC
- Teradata

Lorsqu'une transaction d'insertion, de mise à jour ou de suppression entière est transmise à la source de données pour traitement, le serveur fédéré suppose que la source de données conserve l'atomicité des instructions en cas d'erreur. Lorsque seule une partie de la transaction d'insertion, de mise à jour ou de suppression est transmise à la source de données pour traitement, l'intégralité de la transaction est annulée lorsqu'une erreur se produit.

Modification de données dans un système fédéré

Vous pouvez modifier des données dans un système fédéré en insérant, en mettant à jour et en supprimant des données dans des objets de source de données.

Insertion de données dans des objets de source de données

Pour insérer des données dans des sources de données, utilisez les pseudonymes des objets de source de données dans l'instruction INSERT.

Pour insérer des données à l'aide d'un pseudonyme, tous les privilèges suivants doivent être vérifiés :

- Les privilèges détenus par l'ID autorisation de l'instruction doivent inclure le privilège INSERT sur le pseudonyme (pour que le serveur fédéré accepte la requête).
- L'ID utilisateur sur la source de données doit avoir le privilège INSERT sur l'objet de table sous-jacent (pour que la source de données accepte la requête).
- L'ID utilisateur sur la source de données doit être mappé vers l'ID autorisation sur le serveur fédéré via un mappage utilisateur.

Restrictions

La fédération ne prend pas en charge les opérations INSERT avec des sources de données non relationnelles.

Procédure

Pour insérer des données dans des objets de source de données, émettez l'instruction INSERT.

Exemple : Une table Informix se compose de deux colonnes. La première colonne contient des données INTEGER et la deuxième des données VARCHAR (jusqu'à 20 caractères). Le pseudonyme *infx_table_nn* est enregistré avec le serveur fédéré pour la table Informix.

Vous pouvez émettre des instructions INSERT, UPDATE et DELETE sur la table Informix à l'aide du pseudonyme *infx_table_nn*. L'instruction suivante insère une nouvelle ligne d'informations dans la table Informix :

```
INSERT INTO db2user1.infx_table_nn VALUES(1,'Walter')
```

Mise à jour des données dans les sources de données

Pour mettre à jour les données dans les sources de données, utilisez des pseudonymes pour les objets de source de données dans l'instruction UPDATE.

Avant de commencer

Pour mettre à jour des données à l'aide d'un pseudonyme, tous les privilèges suivants doivent être vérifiés :

- Les privilèges détenus par l'ID autorisation de l'instruction doivent inclure le privilège UPDATE sur le pseudonyme (pour que le serveur fédéré accepte la requête).
- L'ID utilisateur sur la source de données doit avoir le privilège UPDATE sur l'objet de table sous-jacent (pour que la source de données accepte la requête).
- L'ID utilisateur sur la source de données doit être mappé vers l'ID autorisation sur le serveur fédéré via un mappage utilisateur.

Restrictions

La fédération ne prend pas en charge les opérations UPDATE avec certaines sources de données, voir «Restrictions INSERT, UPDATE et DELETE du système fédéré», à la page 140.

Pour mettre à jour des données dans des objets de source de données, émettez l'instruction UPDATE.

Exemple : Une table Informix se compose de deux colonnes. La première colonne contient des données INTEGER et la deuxième des données VARCHAR (jusqu'à 20 caractères). Le pseudonyme *infx_table_nn* est enregistré avec le serveur fédéré pour la table Informix.

Vous pouvez émettre des instructions INSERT, UPDATE et DELETE sur la table Informix à l'aide du pseudonyme *infx_table_nn*. L'instruction suivante met à jour une ligne d'informations dans la table Informix :

```
UPDATE db2user1.infx_table_nn SET c2='Bill' WHERE c1=2
```

Suppression de données des objets de source de données

Pour supprimer des données des sources de données, utilisez les pseudonymes pour les objets de source de données dans l'instruction DELETE.

Avant de commencer

Pour supprimer des données à l'aide d'un pseudonyme, tous les privilèges suivants doivent être vérifiés :

- Les privilèges détenus par l'ID autorisation de l'instruction doivent inclure le privilège DELETE sur le pseudonyme (pour que le serveur fédéré accepte la requête).
- L'ID utilisateur sur la source de données doit avoir le privilège DELETE sur l'objet de table sous-jacent (pour que la source de données accepte la requête).
- L'ID utilisateur sur la source de données doit être mappé vers l'ID autorisation sur le serveur fédéré via un mappage utilisateur.

Restrictions

La fédération ne prend pas en charge les opérations DELETE avec certaines sources de données, voir «Restrictions INSERT, UPDATE et DELETE du système fédéré», à la page 140.

Procédure

Pour supprimer des données des objets de source de données, émettez l'instruction DELETE.

Exemple : Une table Informix se compose de deux colonnes. La première colonne contient des données INTEGER et la deuxième des données VARCHAR (jusqu'à 20 caractères). Le pseudonyme *infx_table_nn* est enregistré avec le serveur fédéré pour la table Informix.

Vous pouvez émettre des instructions INSERT, UPDATE et DELETE sur la table Informix à l'aide du pseudonyme *infx_table_nn*. L'instruction suivante supprime une ligne d'informations de la table Informix :

```
DELETE FROM infx_table_nn WHERE c1=3
```

Sémantiques d'attribution dans un système fédéré

Lorsque vous attribuez des données à une colonne de pseudonyme, le type de données peut changer en fonction des règles d'attribution utilisées par le système fédéré. Vous devez maîtriser les règles d'attribution afin d'obtenir les résultats souhaités.

Les règles permettant de déterminer le type de données cible d'une attribution à une colonne de pseudonyme sont les suivantes :

- Déterminez le type de source local : Le type de source local est déterminé par le type de colonne locale ou le type de résultat local des expressions. Si la source est constante, le type de source local est identique au type de la constante.
- Déterminez le type de cible :
 - Si la source d'attribution n'a pas de source, telles des marqueurs de paramètre et des valeurs NULL, alors le type cible est MIN (type_cible_local, type_cible_éloigné), où type_cible_local correspond au type de données local de la colonne mise à jour et type_cible_éloigné correspond au type de données de la source de données de la colonne mise à jour. type_cible_éloigné fait référence au mappage de type avant par défaut du type de données de la colonne cible éloignée.
 - Si la source d'attribution n'est pas NULL ou marqueurs de paramètre, le type cible est MIN(type_cible_local, type_cible_éloigné, type_source_local).

Définition de MIN(type1, type2)

- Type1 et type2 ne sont pas exactement identiques.
- MIN(type1,type2) = MIN(type2, type1)
- MIN(type1, type2) = type_cible_éloigné(type_cible_local), lorsque MIN(type1, type2) = DECIMAL(0,0)
- BLOB est compatible uniquement avec BLOB, donc MIN(BLOB(x), BLOB(y))=BLOB(z) où z=min(x,y)
- Les types de données TIME et DATE ne sont pas compatibles.
- Les types de date/heure et les chaînes de caractères sont compatibles.
- Dans les bases de données Unicode, les chaînes de caractères et les chaînes graphiques sont compatibles.

Les tableaux suivants indiquent les valeurs minimales pour les types de données de date et d'heure, de chaîne graphique, de chaîne de caractères et numérique.

Tableau 10. Types de données numériques

type1	type2	MIN(type1, type2)
SMALLINT	SMALLINT ou INTEGER ou BIGINT ou REAL ou DOUBLE	SMALLINT
INTEGER	BIGINT ou REAL ou DOUBLE	INTEGER
BIGINT	REAL ou DOUBLE	BIGINT
REAL	DOUBLE	REAL
DECIMAL(w,x)	SMALLINT	DECIMAL(p,0) où $p=w-x$, si $p<5$; SMALLINT, dans les cas contraires
DECIMAL(w,x)	INTEGER	DECIMAL(p,0) où $p=w-x$, si $p<11$; INTEGER, dans les cas contraires
DECIMAL(w,x)	BIGINT	DECIMAL(p,0) où $p=w-x$, si $p<19$; BIGINT, dans les cas contraires
DECIMAL(w,x)	DECIMAL(y,z)	DECIMAL(p,s) où $p=\min(w,y)+\min(w-x,y-z)$, $s=\min(x,z)$
DECIMAL(w,x)	DOUBLE ou REAL	DECIMAL(w,x)

Le tableau suivant répertorie les valeurs minimales des deux types de données pour les types de données de chaîne de caractères.

Tableau 11. Types de données de chaîne de caractères

type1	type2	MIN(type1, type2)
CHAR(x)	CHAR(y) ou VARCHAR(y) ou LONG VARCHAR ou CLOB(y)	CHAR(z) où $z=\min(x,y)$
VARCHAR(x)	VARCHAR(y) ou LONG VARCHAR ou CLOB(y)	VARCHAR(z) où $z=\min(x,y)$
LONG VARCHAR,	CLOB(y)	LONG VARCHAR où $x>32700$, CLOB(x) où $x\leq 32700$
CLOB(x)	CLOB(y)	CLOB(z) où $z=\min(x,y)$

Le tableau suivant répertorie les valeurs minimales des deux types de données pour les types de données de chaîne graphique.

Tableau 12. Types de données de chaîne graphique

type1	type2	MIN(type1, type2)
GRAPHIC(x)	GRAPHIC(y) ou VARGRAPHIC(y) ou LONG VARGRAPHIC ou DBCLOB(y)	GRAPHIC(z) où $z=\min(x,y)$
VARGRAPHIC(x)	VARGRAPHIC(y) ou LONG VARGRAPHIC ou DBCLOB(y)	VARGRAPHIC(z) où $z=\min(x,y)$

Tableau 12. Types de données de chaîne graphique (suite)

type1	type2	MIN(type1, type2)
LONG VARGRAPHIC	DBCLOB(y)	LONG VARGRAPHIC où $x > 32700$, DBCLOB(x) où $x \leq 32700$
DBCLOB(x)	DBCLOB(y)	DBCLOB(z) où $z = \min(x, y)$

Le tableau suivant répertorie les valeurs minimales des deux types de données pour les types de données d'heure et de données.

Tableau 13. Type de données de date et d'heure

type1	type2	MIN(type1, type2)
DATE	TIMESTAMP	DATE
TIME	TIMESTAMP	TIME

Si la longueur des données de type de données CHAR que vous insérez est inférieure à la longueur cible, la source de données remplit le reste de la colonne.

Si vous insérez des données de type DATE ou TIME dans une colonne éloignée du type de données TIMESTAMP, la source de données remplit le reste de la colonne.

Sémantiques d'attribution dans un système fédéré - Exemples

Le tableau suivant affiche plusieurs exemples de l'application des sémantiques d'attribution fédérées dans des requêtes selon un type éloigné et local.

Tableau 14. Exemples de sémantiques d'attribution

Type local	Type éloigné	Votre requête	Requête éloignée générée
FLOAT	INTEGER	set c1=123.23	set c1=INTEGER(123.23)
INTEGER	FLOAT	set c1=123.23	set c1=INTEGER(123.23)
FLOAT	INTEGER	set c1=123	set c1=123
CHAR(10)	CHAR(20)	set c1='123'	set c1='123' ('123' a un type VARCHAR(3) et sa longueur est la plus courte)
CHAR(10)	CHAR(20)	set c1=char23col	set c1=CHAR(char23col, 10)
CHAR(10)	CHAR(20)	set c1=expr1	<ul style="list-style-type: none"> set c1=expr1 -- if expr1 returns char(n) $n \leq 10$ set c1=CHAR(expr1, 10) if expr1 returns char(n) $n > 10$
TIMESTAMP	DATE	set c1= current timestamp	set c1=DATE(current timestamp)

Restrictions des sources de données sur des valeurs de types de données

Pour certains types de données, le serveur fédéré supporte une plus large gamme de valeurs que la source de données.

Lorsque vous utilisez ces types de données, employez les valeurs que la seule source de données supporte. Si vous insérez une valeur hors de l'intervalle pris en charge, la source de données la convertit ou renvoie une erreur.

Si vous utilisez un prédicat avec une valeur hors de l'intervalle pris en charge et que ce prédicat est évalué sur la source de données, l'un des résultats suivants peut être obtenu :

- Le prédicat peut renvoyer un résultat différent que le même prédicat sur le serveur fédéré.
- Le prédicat peut entraîner une erreur.

Heures et horodatages avec 24 dans la zone d'heures

Si vous utilisez un prédicat avec une heure ou un horodatage affichant 24 dans la zone d'heures, vous pouvez obtenir une erreur. Pour éviter tout incident, vous pouvez convertir ces heures ou ces horodatages. Vous pouvez employer une instruction semblable à l'instruction UPDATE suivante :

```
UPDATE my_table SET timestamp_col = timestamp_col + 0 SECONDS;
```

Cette mise à jour passe la zone d'heures à 00:00:00 et incrémente la date d'un jour.

Le serveur fédéré permet aux heures et aux horodatages d'afficher 24 dans la zone d'heures. La différence entre un horodatage de 24 heures et un autre de 00 heures tient à leur distinction en cas de comparaison, alors qu'ils sont identiques sur le plan arithmétique.

Exemple

- Le prédicat suivant renvoie la valeur false :
`2008-05-14-24:00:00.000000 = 2008-05-15-00:00:00.000000`
- Le prédicat suivant renvoie la valeur true :
`(2008-05-14-24:00:00.000000 + 0 SECONDS) = (2008-05-15-00:00:00.000000 + 0 SECONDS)`

Certaines sources de données comme Oracle et Sybase n'acceptent pas les heures ou horodatages affichant 24 dans la zone d'heures.

Chaînes vides sur Oracle

Vous pouvez empêcher des incidents en évitant d'utiliser des chaînes vides dans des applications employant des pseudonymes Oracle. A la place, vous pouvez utiliser la valeur NULL ou la chaîne vide ('').

Dans les colonnes VARCHAR d'un serveur fédéré qui n'est pas compatible avec VARCHAR2, une distinction est faite entre la chaîne vide et les valeurs NULL, qui engendre des comportements différents entre les tables locales et les pseudonymes. En revanche, dans les colonnes VARCHAR d'une source de données distante compatible avec VARCHAR2, la chaîne vide et les valeurs NULL sont considérées comme équivalentes. Si une chaîne vide (') est insérée dans une colonne VARCHAR, elle est convertie en valeur NULL.

Dans les exemples suivants, MY_TABLE est une table locale dans la base de données fédérée et MY_NICK est un pseudonyme sur une table distante dans une base de données Oracle :

Exemple 1

Dans cet exemple, une chaîne vide est insérée dans les tables MY_TABLE et MY_NICK, qui contiennent tous deux une colonne NOT NULL nommée NOT_NULL_COL. L'instruction INSERT sur MY_TABLE aboutit, mais

l'instruction INSERT sur MY_NICK échoue avec une erreur, car la chaîne vide est convertie en valeur NULL, qui entre en conflit avec la contrainte NOT NULL.

```
INSERT INTO my_table(not_null_col) VALUES('');  
INSERT INTO my_nick(not_null_col) VALUES('');
```

Exemple 2

Dans cet exemple, les tables fédérées et distantes sont initialement vides. La première instruction SELECT ne renvoie aucune ligne, car le serveur fédéré fait une distinction entre la chaîne vide et les valeurs NULL, mais la deuxième instruction SELECT renvoie une ligne, car lors de l'insertion, la base de données Oracle convertit la chaîne vide en valeur NULL.

```
INSERT INTO my_table(my_col) VALUES('');  
SELECT * FROM my_table WHERE my_col IS NULL;  
INSERT INTO my_nick(my_col) VALUES('');  
SELECT * FROM my_nick WHERE my_col IS NULL;
```

Mises à jour fédérées à l'aide des points de sauvegarde d'une application

Les points de sauvegarde d'une application figurant dans un système fédéré vous permettent de contrôler le travail effectué par un sous-ensemble d'instructions SQL dans une transaction.

Vous pouvez utiliser plusieurs points de sauvegarde dans une transaction unique. Un sous-ensemble d'instructions SQL regroupé sous un point de sauvegarde peut être considéré comme étant une unité. Vous pouvez logiquement diviser une transaction en un niveau unique ou en des niveaux imbriqués d'unités de points de sauvegarde.

Après avoir défini un point de sauvegarde dans votre application, vous pouvez ensuite le publier ou annuler le travail effectué depuis sa définition. Par exemple, si une instruction SQL individuelle échoue dans un point de sauvegarde, l'unité reste intacte dans son ensemble. Vous pouvez soit :

- Annuler toutes les instructions SQL qui ont été exécutées dans le point de sauvegarde en rétablissant ce dernier.
- Publier le point de sauvegarde lorsque vous n'avez plus besoin du système pour gérer le point de sauvegarde.

Vous pouvez effectuer des mises à jour fédérées à l'aide des points de sauvegarde d'une application dans DB2 Database pour Linux, UNIX, et Windows.

Exemples de mises à jour fédérées à l'aide des points de sauvegarde d'une application

Cet exemple de points de sauvegarde d'une application illustre l'effet des opérations d'annulation dans les points de sauvegarde imbriqués.

Exemple : Créez un pseudonyme nommé NN_DEPT pour la table DEPARTMENT du schéma DEPTDATA dans le serveur DB2_SERVER. Le pseudonyme NN_DEPT contient les colonnes DEPT_NO, DEPT_NAME et MANAGER_NO.

```
CREATE NICKNAME NN_DEPT FOR DB2_SERVER.DEPTDATA.DEPARTMENT;
```

Insérez une ligne avant de créer le point de sauvegarde SP1. Insérez une autre ligne avant de créer le point de sauvegarde SP2. Insérez deux lignes supplémentaires avant de créer le point de sauvegarde SP3. Insérez une cinquième ligne.

```
INSERT INTO NN_DEPT VALUES ('A10', 'SALES', 'ADAM');
SAVEPOINT SP1 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('B10', 'MARKETING', 'BRIAN');
SAVEPOINT SP2 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('C10', 'ENGINEERING', 'CINDY');
INSERT INTO NN_DEPT VALUES ('C20', 'TESTING', 'DOUG');
SAVEPOINT SP3 ON ROLLBACK RETAIN CURSORS;
INSERT INTO NN_DEPT VALUES ('D10', 'SUPPORT', 'EMILY');
```

Le pseudonyme NN_DEPT contient désormais cinq lignes affichant les rayons A10, B10, C10, C20 et D10. Vous souhaitez annuler l'insertion de certaines lignes. Les exemples suivants décrivent les résultats de l'annulation de chaque point de sauvegarde :

- ROLLBACK TO SAVEPOINT SP3;
La dernière ligne contenant le rayon D10 ne figure plus dans le pseudonyme NN_DEPT. Les lignes que vous avez insérées avant de créer le point de sauvegarde SP3 (A10, B10, C10, C20) existent toujours dans le pseudonyme NN_DEPT.
- ROLLBACK TO SAVEPOINT SP2;
Les lignes contenant les rayons C10, C20, D10 ne figurent plus dans le pseudonyme NN_DEPT. Les lignes que vous avez insérées avant de créer le point de sauvegarde SP2 (A10, B10) existent toujours dans le pseudonyme NN_DEPT.
- ROLLBACK TO SAVEPOINT SP1;
Les lignes contenant les rayons B10, C10, C20, D10 ne figurent plus dans le pseudonyme NN_DEPT. La ligne que vous avez insérée avant de créer le point de sauvegarde SP1 (A10) existe toujours dans le pseudonyme NN_DEPT.

Restrictions appliquées aux les mises à jour fédérées à l'aide des points de sauvegarde d'une application

Les systèmes fédérés imposent des restrictions sur les opérations de points de sauvegarde.

La prise en charge d'un point de sauvegarde pour les opérations d'écriture sur des données distantes figurant dans des applications fédérées se limite à DB2 Database pour Linux, UNIX et Windows. Les restrictions suivantes s'appliquent à l'objet de la source de données pour lequel le pseudonyme a été défini (comme cible d'une opération d'écriture) :

- Pour la version 9.5, l'objet de la source de données figurant dans une opération de point de sauvegarde peut être un pseudonyme.
- Pour les versions précédentes, l'objet de la source de données figurant dans une opération de point de sauvegarde ne peut être un pseudonyme.

Vous pouvez activer des points de sauvegarde en mode série, l'environnement du traitement à parallélisme massif (MPP - Massively Parallel Processing) et les environnements de validation en une phase fédérée ou en deux phases fédérées. Toute restriction existant dans ces environnements s'applique également aux opérations de points de sauvegarde.

Chapitre 11. Importation et exportation de données pour les pseudonymes

Vous pouvez utiliser la commande `IMPORT` pour importer des données dans un pseudonyme et la commande `EXPORT` pour exporter des données à partir d'une requête qui référence un pseudonyme.

Vous pouvez utiliser les commandes `IMPORT` avec les sources de données suivantes :

- Famille DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

Restrictions pour l'importation des données dans des pseudonymes

Il existe des restrictions de l'utilisation de la commande `IMPORT` pour importer des données dans un pseudonyme.

Les restrictions suivantes s'appliquent lorsque vous utilisez la commande `IMPORT` pour importer les données dans un pseudonyme :

- L'objet éloigné sur lequel le pseudonyme est défini doit être une table. Vous ne pouvez pas importer de données dans un pseudonyme défini sur une vue ou un synonyme.
- Les types de fichier pris en charge sont IXF, ASC et DEL.
- La clause `ALLOW WRITE ACCESS` doit être indiquée. Cette clause appelle le mode d'importation en ligne. La clause `ALLOW WRITE ACCESS` accorde aux applications concurrentes des droits en lecture et en écriture dans la table cible de l'importation.
- Vous ne pouvez pas utiliser le mode `COMMITCOUNT AUTOMATIC` avec des pseudonymes.
- La commande `COMMITCOUNT n` doit être indiquée, *n* étant un nombre valide différent de zéro.
- Seules les opérations `INSERT` et `INSERT_UPDATE` sont prises en charge avec les pseudonymes.
- Les types de colonne qui ne sont pas pris en charge avec des pseudonymes sont des objets LOB et des colonnes générées. Pour importer des données LOB dans une table éloignée, la colonne de pseudonyme correspondante doit être un type de données `VARCHAR`.

- Les modificateurs `typefichier` ne sont pas pris en charge avec les pseudonymes :
 - `dldelfiletype`
 - `generatedignore`
 - `generatedmissing`
 - `identityignore`
 - `identitymissing`
 - `indexixf`
 - `indexschema`
 - `lobsinfile`
 - `nodefaults`
 - `no_type_idfiletype`
 - `usedefaults`
- La hiérarchie (tables basées sur un type structuré) n'est pas prise en charge avec des pseudonymes.

Si vous soumettez une commande `IMPORT` qui ne respecte pas ces restrictions, le code d'erreur `SQL -27999N` est renvoyé. Par exemple :

```
SQL27999N L'opération IMPORT demandée dans une cible éloignée (pseudonyme)
ne peut pas être effectuée. Code anomalie = "code_anomalie"
```

Commande `IMPORT` avec des pseudonymes - Exemples

Les exemples illustrent comment importer des données dans les pseudonymes à partir de différents types de fichier.

type de fichier `DEL`

Cet exemple utilise l'option `INSERT_UPDATE` pour importer des données à partir d'un type de fichier `DEL` :

```
IMPORT FROM import_file_1.del OF DEL
ALLOW WRITE ACCESS
COMMITCOUNT 50
INSERT_UPDATE INTO NICKNAME_1;
```

type de fichier `IXF`

Cet exemple utilise l'option `INSERT` pour importer des données à partir d'un type de fichier `IXF` :

```
IMPORT FROM import_file_1.ixf OF IXF
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

type de fichier `ASC`

Cet exemple utilise l'option `INSERT` pour importer des données à partir d'un type de fichier `ASC`. L'exemple inclut le modificateur de fichier `STRIPTBLANKS` permettant de tronquer les espaces de fin dans les données. Le paramètre `METHOD L` indique le début et la fin des numéros de colonne.

```
IMPORT FROM import_file_1.asc OF ASC MODIFIED BY STRIPTBLANKS
METHOD L(1 6, 8 32, 34 44, 46 48)
ALLOW WRITE ACCESS
COMMITCOUNT 20
INSERT INTO NICKNAME_1;
```

Restrictions pour l'exportation de données à l'aide de pseudonymes

Il existe des restrictions de l'utilisation de la commande EXPORT pour exporter des données à partir d'une requête qui référence un pseudonyme.

Les restrictions suivantes s'appliquent lorsque vous utilisez la commande EXPORT pour exporter des données à l'aide d'un pseudonyme :

- La description de la table cible requise pour l'importation de l'opération CREATE n'est pas enregistrée avec le format de fichier IXF. Ayez recours à l'utilitaire db2look pour rassembler les informations requises pour la recréation de la table.
- Vous pouvez exporter des données dans les types de fichier IXF et DEL. Le type de fichier ASC n'est pas pris en charge pour l'exportation de données à partir de pseudonymes.

Chapitre 12. Utilisation de pseudonymes

Un pseudonyme est un identificateur qu'une application utilise afin de référencer un objet de source de données, comme une table ou une vue. Dans un système fédéré, vous pouvez employer des pseudonymes afin d'accéder à des objets de source de données et améliorer les performances des requêtes sur des sources de données éloignées.

Pseudonyme dans un système fédéré

Lorsque vous souhaitez sélectionner ou modifier des données de la source de données, vous interrogez les pseudonymes à l'aide des instructions SELECT, INSERT, UPDATE et DELETE. Vous soumettez des requêtes dans DB2 SQL sur la base de données fédérée.

Vous pouvez joindre des données à partir de tables locales ou de sources de données éloignées avec une instruction SQL, comme si toutes les données étaient locales. Par exemple, vous pouvez joindre des données qui se trouvent dans les objets suivants :

- une table DB2 locale dans la base de données fédérée, une table Oracle et une vue Sybase
- une table DB2 UDB pour z/OS sur un serveur, une table DB2 UDB for z/OS sur un autre serveur et une feuille de calcul Excel

En traitant des instructions SQL comme si les sources de données étaient des vues ou des tables relationnelles ordinaires dans la base de données fédérée, le système fédéré peut joindre des données relationnelles avec des données à des formats non relationnels.

Les tables et les vues dans la base de données fédérée sont des *objets locaux*. Ne créez pas de pseudonymes pour ces objets. A la place, utilisez le nom d'objet réel dans les instructions SQL.

Les *objets éloignés* sont des objets qui ne se trouvent pas dans la base de données fédérée. Vous devez créer des pseudonymes pour ces objets. Par exemple :

- Tables et vues d'une autre base de données ou d'une instance sur le système fédéré
- Tables et vues d'une autre base de données ou d'une instance sur un autre système
- Tables et vues dans des sources de données, Oracle, Sybase et ODBC, par exemple.

Curseurs déclarés WITH HOLD

Vous pouvez utiliser l'option WITH HOLD sur un curseur défini sur un pseudonyme.

Pour l'encapsuleur DRDA et la source de données DB2 Database for Linux, UNIX et Windows, les curseurs déclarés avec l'option WITH HOLD restent ouverts sur plusieurs unités de travail.

Si vous déclarez un curseur WITH HOLD et que la source de données ne supporte pas cette option, l'attribut est ignoré.

Déclencheurs

Un pseudonyme ne peut pas être une cible de mise à jour dans un déclencheur.

Vous pouvez inclure des instructions SELECT pour des pseudonymes dans le corps du déclencheur. Vous ne pouvez pas inclure d'instructions INSERT, UPDATE ou DELETE pour les pseudonymes dans le corps du déclencheur.

Accès aux données avec des pseudonymes

Avec un système fédéré, vous pouvez facilement accéder aux données, quel que soit l'emplacement de stockage. Pour accéder aux données, vous créez des pseudonymes pour les objets source de données, tels que des tables et des vues.

Par exemple, si le pseudonyme DEPT représente la table éloignée EUROPE.PERSON.DEPT, vous pouvez utiliser l'instruction SELECT * FROM DEPT pour interroger les informations dans la table éloignée. Lorsque vous interrogez un pseudonyme, il n'est pas nécessaire de se souvenir des détails de connexion sur la source de données. La prise en compte de ces éléments n'est donc pas obligatoire :

- Nom de l'objet sur la source de données
- Serveur sur lequel se trouve l'objet de source de données
- Type de source de données sur lequel se trouve l'objet, tel qu'Informix et Oracle
- Langage d'interrogation ou dialecte SQL utilisé par la source de données
- Mappages de type de données entre la source de données et le serveur fédéré
- Mappages de fonctions entre la source de données et le serveur fédéré

Les métadonnées sous-jacentes du catalogue de la base de données fédérée fournissent au serveur fédéré les informations requises pour le traitement des requêtes. Ces métadonnées sont rassemblées à partir des sources de données lorsque le serveur fédéré et la base de données sont configurés pour l'accès aux sources de données.

Une fois le système fédéré configuré, vous utilisez les pseudonymes pour interroger les sources de données ou pour améliorer la configuration du système fédéré.

Instructions SQL pouvant être utilisées avec des pseudonymes

Les instructions SQL prennent en charge l'utilisation de pseudonymes.

Tableau 15. Instructions SQL communes pour utilisation avec des pseudonymes

Instruction SQL	Description	Autorisation requise
ALTER NICKNAME	Modifie un pseudonyme existant en changeant le nom de la colonne locale, le type de données local, les options de colonne fédérées ou les contraintes informationnelles. La table ou la vue sur la source de données n'est pas affectée.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège ALTER ou CONTROL sur le pseudonyme • privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe • Créateur du pseudonyme, tel qu'il est enregistré dans la colonne DEFINER de la vue de catalogue du pseudonyme
ALTER TABLE	Change une table éloignée qui a été créée via la base de données fédérée en utilisant DDL transparent. Vous ne pouvez pas modifier les tables créées de manière native sur la source de données. Peut utiliser des contraintes référentielles pour ajouter une contrainte d'intégrité référentielle à un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège ALTER ou CONTROL sur le pseudonyme • privilège ALTERIN sur le schéma, si le nom de schéma du pseudonyme existe
COMMENT ON	Ajoute ou remplace les commentaires dans les descriptions de catalogue des différents objets, incluant les fonctions, les mappages de fonctions, les index, les pseudonymes, les serveurs, les options de serveur, les mappages de type et les encapsuleurs.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège ALTER ou CONTROL sur l'objet • Privilège ALTERIN sur le schéma • Créateur de l'objet, tel qu'il est enregistré dans la colonne DEFINER de la vue de catalogue de l'objet
CREATE ALIAS	Définit un pseudonyme pour un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Droit IMPLICIT_SCHEMA sur la base de données, si le nom de schéma implicite ou explicite de le pseudonyme n'existe pas • Droit CREATEIN sur le schéma, si le nom de schéma du pseudonyme fait référence à un schéma existant
Index CREATE INDEX avec la clause SPECIFICATION ONLY	Crée une spécification d'index (métadonnées) qui indique à l'optimiseur de requêtes qu'un objet de source de données a un index. Aucun index n'a été créé, seule la spécification est créée.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège CONTROL ou INDEX sur l'objet de source de données sous-jacent — soit droit IMPLICIT_SCHEMA sur la base de données, soit privilège CREATEIN sur le schéma

Tableau 15. Instructions SQL communes pour utilisation avec des pseudonymes (suite)

Instruction SQL	Description	Autorisation requise
CREATE TABLE avec la clause OPTIONS	Crée une table éloignée via la base de données fédérée utilisant DDL transparent.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège CREATETAB sur la base de données et privilège USE sur l'espace de table —<i>et soit</i> droit d'accès IMPLICIT_SCHEMA à la base de données, soit droit d'accès CREATEIN à le schéma
CREATE TABLE avec instruction AS fullselect et clauses DATA INITIALLY DEFERRED REFRESH	Crée une table de requêtes matérialisées à l'aide d'une instruction fullselect qui référence un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège CREATETAB sur la base de données et privilège USE sur l'espace de table —<i>et soit</i> droit d'accès IMPLICIT_SCHEMA à la base de données, soit droit d'accès CREATEIN à le schéma • Privilège CONTROL sur la table ou la vue • Droit SELECT sur la table ou la vue et ALTER si REFRESH DEFERRED est indiqué
CREATE VIEW	Crée une vue qui référence un ou plusieurs pseudonymes.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège CONTROL ou SELECT sur le pseudonyme —<i>et soit</i> droit IMPLICIT_SCHEMA sur la base de données, soit privilège CREATEIN sur le schéma
DELETE	Supprime des lignes de l'objet de source de données, tel une table ou une vue ayant un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège DELETE sur le pseudonyme et privilège DELETE sur l'objet de source de données sous-jacent • Privilège CONTROL sur l'objet de source de données sous-jacent
DROP	Supprime un objet, tel un pseudonyme, une vue fédérée ou une spécification d'index. La table, la vue ou l'index sur la source de données n'est pas affectée. Lorsque les tables créées à l'aide de DDL transparent sont supprimées, le pseudonyme correspondant de cette table est également supprimé.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège DROPIN sur le schéma pour l'objet • Privilège CONTROL sur l'objet

Tableau 15. Instructions SQL communes pour utilisation avec des pseudonymes (suite)

Instruction SQL	Description	Autorisation requise
GRANT	Accorde des privilèges aux pseudonymes et vues fédérés (ALTER, DELETE, INDEX, INSERT, SELECT ou UPDATE, par exemple). Les privilèges sur la source de données doivent être accordés séparément.	<ul style="list-style-type: none"> • SYSADM ou DBADM • WITH GRANT OPTION pour chaque privilège identifié • Privilège CONTROL sur l'objet
INSERT	Insère des lignes dans l'objet de source de données, tel une table ou une vue ayant un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège INSERT sur le pseudonyme et privilège INSERT sur l'objet de source de données sous-jacent • Privilège CONTROL sur l'objet de source de données sous-jacent
LOCK TABLE	Provoque le verrouillage de l'objet éloigné sur la source de données. Empêche les traitements d'application simultanés de changer une source de données ayant un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège SELECT sur la table sous-jacente • Privilège CONTROL sur la table sous-jacente.
REVOKE	Révoque des privilèges sur des pseudonymes et des vues fédérées (ALTER, DELETE, INDEX, INSERT, SELECT ou UPDATE, par exemple). Les privilèges sur la source de données doivent être révoqués séparément.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège CONTROL sur l'objet
SELECT	Sélectionne des lignes de l'objet de source de données, tel une table ou une vue ayant un pseudonyme.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège SELECT sur le pseudonyme et privilège SELECT sur l'objet de source de données sous-jacent • Privilège CONTROL sur l'objet de source de données sous-jacent
UPDATE	Met à jour les valeurs dans des lignes de colonnes de l'objet de source de données, une table ou une vue ayant un pseudonyme, par exemple.	<ul style="list-style-type: none"> • SYSADM ou DBADM • Privilège UPDATE sur le pseudonyme et privilège UPDATE sur l'objet de source de données sous-jacent • Privilège CONTROL sur l'objet de source de données sous-jacent

Lorsque vous soumettez une requête à la base de données fédérée, les privilèges d'autorisation pour le pseudonyme de la requête sont vérifiés. Les autorisations requises pour l'objet de source de données auquel fait référence le pseudonyme sont appliquées uniquement lorsque la requête est traitée sur la source de données.

Pour sélectionner, insérer, mettre à jour ou supprimer des données avec un pseudonyme, l’ID autorisation de l’instruction doit inclure ces privilèges :

- Le privilège approprié sur le pseudonyme de la base de données fédérée doit accepter la requête.
- Privilège approprié sur l’objet de table sous-jacent de la source de données permettant d’accepter la requête

Par exemple, pour mettre à jour une source de données à l’aide d’un pseudonyme, le privilège UPDATE sur l’objet et le privilège UPDATE sur la source de données sous-jacente sont requis.

Accès aux nouveaux objets de source de données

Pour accéder à de nouveaux objets de source de données, vous devez créer des pseudonymes. Utilisez l’instruction CREATE NICKNAME pour les sources de données n’ayant pas de pseudonyme.

Avant de commencer

Le système fédéré doit être configuré pour accéder à la source de données.

Il doit exister une définition de serveur pour le serveur de source de données sur lequel se trouve l’objet dans la base de données fédérée. Vous créez une définition de serveur avec l’instruction CREATE SERVER.

Pour insérer, mettre à jour ou supprimer des données avec un pseudonyme, l’ensemble des privilèges suivants doivent être respectés :

- Les privilèges détenus par l’ID autorisation de l’instruction doivent inclure les privilèges SELECT, INSERT, UPDATE et DELETE sur le pseudonyme pour que la base de données fédérée accepte la requête.
- L’ID utilisateur sur la source de données doit disposer des privilèges SELECT, INSERT, UPDATE et DELETE sur l’objet de table sous-jacent pour que la source de données accepte la requête.
- L’ID utilisateur sur la source de données doit être mappé vers l’ID autorisation sur le serveur fédéré via un mappage utilisateur.

Vous devez disposer des autorisations requises pour l’instruction CREATE NICKNAME.

- SYSADM ou DBADM
- Droit IMPLICIT_SCHEMA sur la base de données fédérée, si le nom de schéma explicite ou implicite du pseudonyme n’existe pas
- Privilège CREATEIN sur le schéma, si le nom de schéma du pseudonyme existe

A propos de cette tâche

Vous devez accéder régulièrement aux objets de source de données n’ayant pas de pseudonymes. De nouveaux objets peuvent avoir été ajouté à une source de données, une vue nouvellement créée, par exemple. Il peut s’agir d’objets existants qui n’avaient pas été enregistrés avec le serveur fédéré lors de la configuration. Vous devez alors créer un pseudonyme pour l’objet.

Procédure

Pour accéder aux nouveaux objets de source de données, émettez l'instruction CREATE NICKNAME.

Création de pseudonymes pour les sources de données relationnelles et non relationnelles

L'instruction CREATE NICKNAME est légèrement différente pour les sources de données relationnelles et non relationnelles.

Pour les sources de données relationnelles, la syntaxe de l'instruction CREATE NICKNAME est la suivante :

```
CREATE NICKNAME nom_pseudonyme FOR nom_serveur."schéma_distant"."nom_objet"
```

nickname_name

Pseudonyme unique de l'objet de source de données. La longueur des pseudonymes peut aller jusqu'à 128 caractères.

Le pseudonyme est un nom composé de deux parties, le schéma et le pseudonyme. Si vous omettez le schéma lors de la création du pseudonyme, le schéma du pseudonyme sera l'ID authentification de l'utilisateur créant le pseudonyme. Les valeurs par défaut du schéma sont choisies en fonction de la hiérarchie suivante :

1. Registre spécial CURRENT SCHEMA
2. Registre spécial SESSION_USER
3. Registre spécial SYSTEM USER
4. ID autorisation connecté à la base de données

```
FOR nom_serveur."schéma_distant"."nom_objet"
```

Identificateur en trois parties pour l'objet de source de données éloigné. Si la source de données ne prend pas en charge les schémas, omettez le schéma de l'instruction CREATE NICKNAME.

- *nom_serveur* correspond au nom attribué au serveur de source de données dans l'instruction CREATE SERVER.
- *schéma_éloigné* correspond au nom du schéma éloigné auquel l'objet appartient.
- *nom_objet* correspond au nom de l'objet éloigné auquel vous souhaitez accéder.

OPTIONS (*liste_options*)

Informations sur le pseudonyme qui activent le compilateur de requêtes SQL et sur l'encapsuleur afin d'exécuter de manière efficace les requêtes.

Pour certaines sources de données non relationnelles, la syntaxe de l'instruction CREATE NICKNAME est la suivante :

```
CREATE  
NICKNAME nom_pseudonyme liste_définition_colonnes  
FOR SERVER nom_serveur  
OPTIONS (liste_options)
```

nickname_name

Pseudonyme unique pour l'objet de source de données, comme il est décrit ci-dessus pour les sources de données relationnelles.

liste_définitions_colonne

Liste de colonnes de pseudonyme et de types de données.

FOR SERVER *nom_serveur*

Nom local créé pour le serveur éloigné dans l'instruction CREATE SERVER des informations de définition de serveur.

OPTIONS (*liste_options*)

Informations sur le pseudonyme qui activent le compilateur de requêtes SQL et sur l'encapsuleur afin d'exécuter de manière efficace les requêtes.

Noms d'index et de colonnes de pseudonymes

Lorsque vous créez un pseudonyme, le serveur fédéré utilise un algorithme pour générer des noms d'index et de colonnes de pseudonymes.

Cet algorithme est amélioré dans la version 9.5 pour que les noms d'index et de colonnes de pseudonymes correspondent le plus possible aux noms d'origine.

L'algorithme s'applique uniquement aux sources de données relationnelles. Pour les sources de données non relationnelles, le nom indiqué dans l'instruction CREATE NICKNAME reste le même.

Le tableau suivant montre des exemples de noms de colonnes de pseudonymes à partir des noms de colonnes éloignées. Dans ces exemples, les colonnes éloignées se trouvent dans des tables éloignées distinctes. Les noms de colonnes de pseudonymes différents des noms de colonnes éloignées apparaissent en gras.

Tableau 16. Noms de colonnes de pseudonymes générés à partir des noms de colonnes éloignées

Nom de colonne éloignée	Nom obtenu avec l'algorithme en cours pour des sources de données qui...			Nom obtenu avant la version 9.5 (l'algorithme compacte et convertit les caractères non alpha-numériques)
	Compactent des identificateurs en majuscules	Compactent des identificateurs en minuscules	Laissent les identificateurs inchangés	
column1	column1	COLUMN1	COLUMN1	COLUMN1
Column1	Column1	Column1	COLUMN1	COLUMN1
COLUMN1	COLUMN1	COLUMN1	COLUMN1	COLUMN1
column-1	column-1	column-1	column-1	COLUMN_1
Column-1	Column-1	Column-1	Column-1	COLUMN_1
COLUMN-1	COLUMN-1	COLUMN-1	COLUMN-1	COLUMN_1

Si les colonnes éloignées se trouvent dans la même table, le serveur fédéré utilise l'algorithme existant qui génère des noms uniques. Par exemple, pour les noms de colonnes de pseudonymes pour les sources de données qui laissent les identificateurs inchangés, les noms de colonnes éloignées sont les suivants :

- column1 devient COLUMN1
- Column1 devient COLUMN10
- COLUMN1 devient COLUMN11

Accès aux sources de données à l'aide de sessions passe-système

Vous pouvez soumettre des instructions SQL directement aux sources de données à l'aide d'un mode spécial appelé passe-système. La session passe-système permet de soumettre des instructions SQL dans le dialecte SQL pour cette source de données.

Utilisez une session passe-système pour exécuter une opération qui n'est pas réalisable à l'aide de SQL. Par exemple, utilisez une session passe-système pour créer une procédure ou un index ou effectuez des requêtes dans le dialecte natif de la source de données.

Les sources de données qui prennent en charge le passe-système uniquement acceptent les instructions SQL dans une session passe-système.

De la même manière, vous pouvez utiliser une session passe-système pour effectuer des actions qui ne sont pas prises en charge par SQL, comme certaines tâches d'administration. Les tâches d'administration que vous pouvez effectuer dépendent de la source de données. Par exemple, pour DB2 Database pour Linux, UNIX et Windows, vous pouvez exécuter l'utilitaire des statistiques pour la source de données mais vous ne pouvez pas démarrer ou arrêter la base de données éloignée.

Vous pouvez interroger une seule source de données à la fois dans une session passe-système. Utilisez la commande SET PASSTHRU pour ouvrir une session. Lorsque vous utilisez la commande SET PASSTHRU RESET, la session passe-système se ferme. Si vous utilisez la commande SET PASSTHRU et non la commande SET PASSTHRU RESET, la session passe-système en cours est fermée et une nouvelle session passe-système s'ouvre.

Vous pouvez déclarer un curseur WITH HOLD dans une session passe-système. Si vous utilisez cette option avec l'encapsuleur DRDA et la source de données DB2 Database for Linux, UNIX et Windows, les curseurs déclarés restent ouverts sur toutes les unités de travail. Si vous déclarez un curseur WITH HOLD et que la source de données ne supporte pas cette option, l'attribut est ignoré.

Vous ne pouvez pas utiliser de sessions passe-système avec des sources de données non relationnelles.

Accès aux données hétérogènes via des vues fédérées

Une *vue fédérée* est une vue de la base de données fédérée dont les tables de base se trouvent sur des sources de données éloignées. La vue fédérée référence les tables de base avec des pseudonymes, et non avec des noms de table de source de données.

Avant de commencer

Vous devez disposer d'une des autorisations suivantes pour émettre l'instruction CREATE VIEW :

- SYSADM ou DBADM
- Pour chaque pseudonyme d'une instruction fullselect :
 - Privilège CONTROL ou SELECT sur la table ou la vue sous-jacente
 - Un des droits ou privilèges suivants :
 - Droit IMPLICIT_SCHEMA sur la base de données fédérée, si le nom de schéma explicite ou implicite de la vue n'existe pas

- Privilège CREATEIN sur le schéma, si le nom de schéma de la vue fait référence à un schéma existant.

Les privilèges des objets sous-jacents ne sont pas pris en compte lors de la définition d'une vue sur un pseudonyme de base de données fédérée.

Restrictions

Les vues fédérées avec des instructions UNION sont des vues en lecture seule.

Les vues fédérées qui incluent plus d'un pseudonyme dans la clause FROM sont des vues en lecture seule.

Les vues fédérées qui incluent un seul pseudonyme dans la clause FROM peuvent être des vues en lecture seule.

- Si le pseudonyme de la clause FROM concerne une source de données non relationnelle, la vue fédérée est en lecture seule.
- Si vous incluez d'autres pseudonymes en tant que prédicats ou en tant que sous-requêtes lorsque vous créez la vue, la vue fédérée peut être mise à jour.

A propos de cette tâche

Lorsque vous interrogez une vue fédérée, les données sont extraites de la source de données éloignée. L'action de création d'une vue de base de données fédérée d'une source de données est parfois appelée «création d'une vue sur un pseudonyme.» En effet, vous référencez les pseudonymes et non les sources de données lorsque vous créez la vue.

Ces vues offrent un niveau d'indépendance de données élevé pour une base de données globalement intégrée, de la même manière que les vues définies sur plusieurs tables locales pour les gestionnaires de bases de données relationnels centralisés.

Procédure

Pour créer une vue fédérée, émettez l'instruction CREATE VIEW.

Les autorisations requises pour la source de données de la table ou de la vue référencée par le pseudonyme sont appliquées lors du traitement de la requête. L'ID autorisation de l'instruction peut être mappé vers un différent ID éloigné par un mappage utilisateur.

Création de vues fédérées - Exemples

Ces exemples indiquent comment créer des vues fédérées pour accéder aux données à partir de plusieurs sources de données. Ces exemples présentent la syntaxe de l'instruction CREATE VIEW pour la fédération.

Exemple : Création d'une vue fédérée qui fusionne des données similaires provenant de plusieurs objets de source de données

Vous utilisez des données client sur plusieurs serveurs : un en Europe, un en Asie et un autre en Amérique du Sud. Une table Oracle correspond aux clients Europe. Le pseudonyme de cette table est ORA_EU_CUST. Une table Sybase correspond aux clients Asie. Le pseudonyme de cette table est SYB_AS_CUST. Une table Informix correspond aux clients Amérique du Sud. Le pseudonyme de cette table est INFMX_SA_CUST. Chaque table a des colonnes contenant le numéro client (CUST_NO), le nom du client (CUST_NAME), le numéro du produit (PROD_NO) et la quantité commandée (QUANTITY). Pour créer une vue à partir des pseudonymes qui fusionnent les données clients, émettez l'instruction suivante :

```
CREATE VIEW FV1
AS SELECT * FROM ORA_EU_CUST
UNION
SELECT * FROM SYB_AS_CUST
UNION
SELECT * FROM INFMX_SA_CUST
```

Exemple : Jointure de données pour la création d'une vue fédérée

Vous utilisez les données client sur un serveur et les données de ventes sur un autre serveur. Les données client se trouvent dans une table Oracle. Le pseudonyme de cette table est ORA_EU_CUST. Les données de ventes se trouvent dans une table Sybase. Le pseudonyme de cette table est SYB_SALES. Vous souhaitez faire correspondre les informations client avec les achats effectués par ces clients. Chaque table a une colonne contenant le numéro de client (CUST_NO). Pour créer une vue fédérée à partir de ces pseudonymes qui joint ces données, émettez l'instruction suivante :

```
CREATE VIEW FV4
AS SELECT A.CUST_NO, A.CUST_NAME, B.PROD_NO, B.QUANTITY
FROM ORA_EU_CUST A, SYB_SALES B
WHERE A.CUST_NO=B.CUST_NO
```

Création d'un pseudonyme sur un pseudonyme

Vous pouvez créer un pseudonyme sur un pseudonyme.

Procédure

Pour créer un pseudonyme sur un pseudonyme, suivez la procédure de cet exemple.

Exemple : Accès à une feuille de calcul Microsoft Excel à partir d'un serveur fédéré AIX

Vous avez un serveur fédéré sur AIX et un serveur fédéré sur Windows. Vous souhaitez accéder à une feuille de calcul Excel à partir des deux serveurs fédérés. Toutefois, l'encapsuleur Excel est pris en charge uniquement sur des serveurs fédérés sur Windows. Pour accéder à la feuille de calcul Excel à partir du serveur fédéré AIX :

1. Sur le serveur fédéré Windows, installez IBM InfoSphere Federation Server.
2. Configurez le serveur fédéré Windows pour accéder aux sources de données Excel.
3. Sur le serveur fédéré Windows, créez un pseudonyme pour la feuille de calcul Excel.

4. Sur le serveur fédéré AIX, installez IBM InfoSphere Federation Server.
5. Configurez le serveur fédéré AIX pour accéder aux sources de données de la famille DB2.
6. Sur le serveur fédéré AIX, créez un pseudonyme Excel sur le serveur fédéré Windows.

Sélection de données dans un système fédéré

La procédure de sélection des données dans un système fédéré dépend du type de données dans lequel vous avez effectué les sélections.

Certains des types de requête répartie utilisés avec un système fédéré sont des requêtes qui interrogent :

- Une seule source de données éloignée
- Une source de données locale et une source de données éloignée
- Plusieurs sources de données éloignées
- Une combinaison de sources de données locales et éloignées

Pour sélectionner des données dans les sources de données, utilisez les pseudonymes pour les objets de source de données dans l'instruction SELECT.

La base de données fédérée est une source de données locale. Les tables et les vues de la base de données fédérée sont des objets locaux. Vous ne créez pas de pseudonymes pour ces objets, vous utilisez le nom d'objet réel dans les instructions SELECT.

Les sources de données éloignées incluent : une autre instance de base de données DB2 Database for Linux, UNIX et Windows sur le serveur fédéré, une autre instance de base de données DB2 Database for Linux, UNIX et Windows sur un autre serveur et d'autres sources données. Les objets qui se trouvent sur des sources de données éloignées sont des objets éloignés.

Sélection de données dans un système fédéré - Exemples

Cette rubrique illustre un scénario dans lequel un serveur fédéré accède à plusieurs sources de données et inclut des exemples d'instruction SELECT.

Exemple : un serveur fédéré est configuré pour accéder à une source de données DB2 for z/OS, une source de données DB2 for System i et une source de données Oracle. Une table contenant des informations de ventes est stockée dans chaque source de données. Cette configuration est décrite dans la figure suivante.

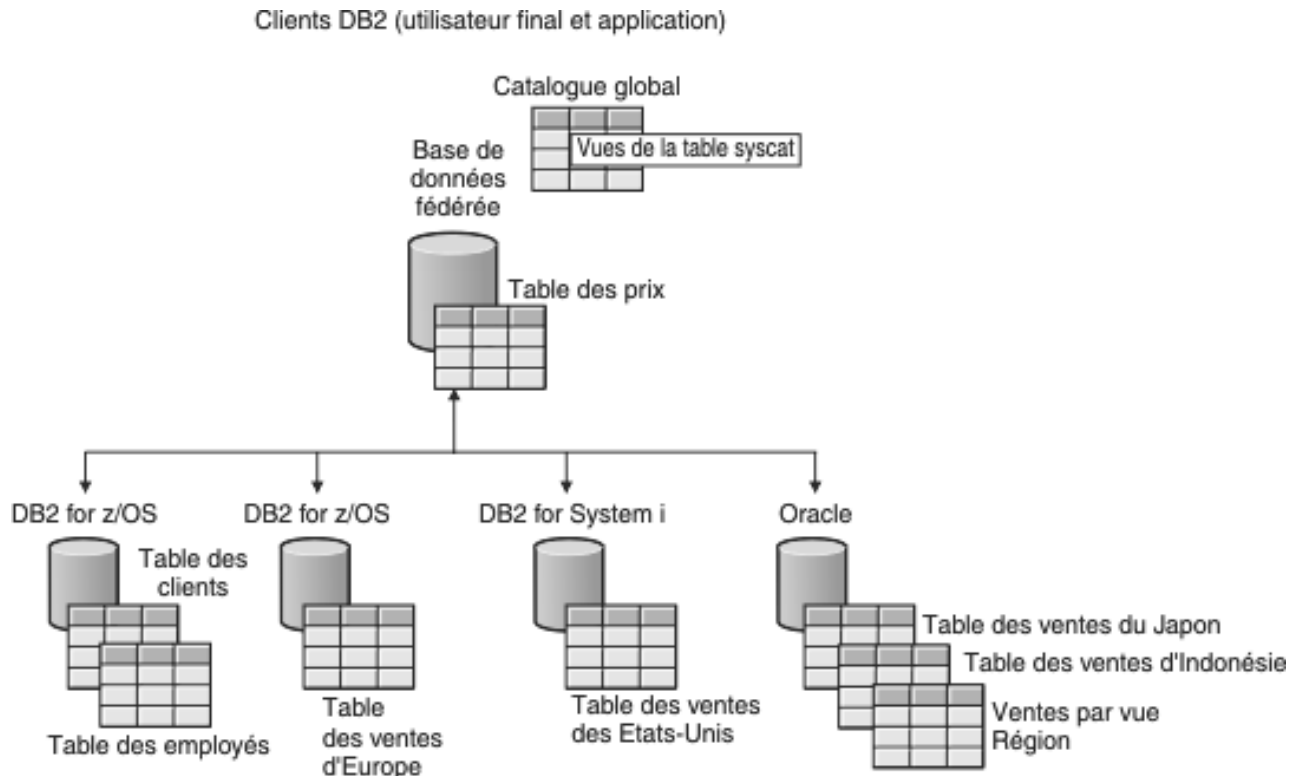


Figure 8. Système fédéré exemple avec sources de données DB2 et Oracle

Les tables de ventes incluent des colonnes qui enregistrent le numéro client (CUST_NO), la quantité commandée (QUANTITY) et le numéro du produit commandé (PROD_NO). De plus, une table locale de la base de données fédérée contient des informations sur les prix. Le tableau des prix inclut des colonnes qui enregistrent le numéro du produit (PROD_NO) et le prix actuel (PRICE).

Les pseudonymes pour les objets de source de données éloignée sont stockés dans les tables SYSCAT.TABLES, comme illustré dans les tables ci-après. La colonne TYPE indique le type d'objet, tel que pseudonyme (N), table locale (T) ou vue (V).

Tableau 17. Informations sur la source de données

Nom de l'objet de source de données	Type d'objet	Emplacement
PRICES	Table locale	Base de données fédérée
EUROPE_SALES	Table éloignée	base de données DB2 for z/OS
US_SALES	Table éloignée	Base de données DB2 for System i
JAPAN_SALES	Table éloignée	Base de données Oracle
SALES_BY_REGION	Vue éloignée	Base de données Oracle

Tableau 18. Tables SYSCAT

TABNAME	TYPE
PRICES	T
FED_PRICES	N

Tableau 18. Tables SYSCAT (suite)

TABNAME	TYPE
Z_EU_SALES	N
SI_US_SALES	N
ORA_JAPANSALES	N
ORA_REGIONSALES	N
...	

Pour sélectionner des données à l'aide d'un pseudonyme, tous les privilèges suivants doivent être vérifiés :

- Les privilèges détenus par l'ID autorisation de l'instruction doivent inclure le privilège SELECT sur le pseudonyme (pour que le serveur fédéré accepte la requête).
- L'ID utilisateur sur la source de données doit avoir le privilège SELECT sur l'objet de table sous-jacent (pour que la source de données accepte la requête).
- L'ID utilisateur sur la source de données doit être mappé vers l'ID autorisation sur le serveur fédéré via un mappage utilisateur.

Les exemples d'instruction SELECT suivants utilisent le système fédéré exemple décrit ci-dessus.

Exemple : Interrogation d'une source de données unique :

Z_EU_SALES contient les produits commandés par les clients européens. Cette table inclut également les quantités commandées à chaque vente. Cette requête utilise une instruction SELECT avec une clause ORDER BY pour répertorier les ventes en Europe et trie la liste en fonction des numéros client :

```
SELECT CUST_NO, PROD_NO, QUANTITY
FROM Z_EU_SALES
ORDER BY CUST_NO
```

Exemple : Jointure d'une base de données locale et d'une source de données éloignée :

PRICES est une table qui se trouve dans la base de données fédérée. Elle contient la liste des prix des produits que vous vendez. Vous souhaitez sélectionner les prix dans cette table locale qui correspondent aux produits répertoriés dans Z_EU_SALES. Vous souhaitez également trier l'ensemble des résultats en fonction du numéro de client.

```
SELECT sales.CUST_NO, sales.PROD_NO, sales.QUANTITY
FROM Z_EU_SALES sales, PRICES
WHERE sales.PROD_NO=PRICES.PROD_NO
ORDER BY sales.CUST_NO
```

Exemple : Interrogation de plusieurs sources de données éloignées :

Vous souhaitez rassembler l'ensemble des informations de vente pour chaque région et organiser l'ensemble de résultats en fonction du numéro de produit.

```
WITH GLOBAL_SALES (Customer, Product, Quantity) AS
(SELECT CUST_NO, PROD_NO, QUANTITY FROM Z_EU_SALES
UNION ALL
SELECT CUST.NO,PROD.NO, QUANTITY FROM IS_US_SALES
UNION ALL
```

```

SELECT CUST.NO,PROD.NO, QUANTITY FROM ORA_JAPANSALES)
SELECT Customer, Product, Quantity
FROM GLOBAL_SALES
ORDER BY Product

```

Vue sur la source de données Oracle qui répertorie les ventes pour le Japon et l'Indonésie. Le pseudonyme de cette vue est ORA_SALESREGION. Vous souhaitez associer ces informations aux ventes des Etats-Unis et afficher les prix des produits à côté de chaque vente.

```

SELECT us_jpn_ind.CUST_NO, us_jpn_ind.PROD_NO,
       us_jpn_ind.QUANTITY, us_jpn_ind.QUANTITY*PRICES.PRICE
AS SALEPRICE FROM
(SELECT CUST_NO, PROD_NO, QUANTITY
FROM ORA_SALESREGION
UNION ALL
SELECT CUST_NO, PROD_NO, QUANTITY
FROM iS_US_SALES us ) us_jpn_ind,PRICES
WHERE us_jpn_ind.PROD_NO = PRICES.PROD_NO
ORDER BY SALEPRICE DESC

```

Contraintes informationnelles sur les pseudonymes

Vous pouvez utiliser des contraintes informationnelles pour les pseudonymes afin d'améliorer les performances des requêtes. Les contraintes informationnelles sont des règles utilisées par l'optimiseur afin d'améliorer les performances mais le gestionnaire de bases de données ne les applique pas.

Vous pouvez indiquer les éléments suivants pour les pseudonymes :

- Contraintes référentielles
- Contraintes de vérification
- Contraintes de dépendance fonctionnelles
- Contraintes de clé primaire
- Contraintes uniques

Indication des contraintes informationnelles sur les pseudonymes (Centre de contrôle DB2)

Pour améliorer les performances des requêtes sur des sources de données éloignées, vous pouvez spécifier des contraintes informationnelles aux pseudonymes. Toutefois, le serveur fédéré n'applique ou ne vérifie pas les contraintes. Vous pouvez indiquer des contraintes informationnelles sur un pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Restrictions

Après avoir défini les contraintes informationnelles sur un pseudonyme, vous pouvez modifier uniquement les noms de colonne pour ce pseudonyme après avoir supprimé les contraintes informationnelles.

A propos de cette tâche

Pour les sources de données relationnelles, vous pouvez indiquer des contraintes informationnelles lorsque vous modifiez un pseudonyme.

Pour les sources de données non relationnelles, vous pouvez indiquer des contraintes informationnelles lorsque vous créez ou modifiez un pseudonyme.

Procédure

Pour indiquer les contraintes informationnelles sur un pseudonyme depuis le Centre de contrôle DB2 :

1. Sélectionnez le dossier **Pseudonymes** :
 - Si vous créez un pseudonyme, dans le panneau des détails de l'objet du Centre de contrôle DB2, cliquez sur **Création de pseudonymes** dans la liste d'actions. L'assistant Pseudonymes s'affiche.
 - Dans la fenêtre Création de pseudonymes, ouvrez l'assistant Ajout d'un pseudonyme ou le bloc-notes Ajout d'un pseudonyme ou le bloc-notes Propriétés pour un pseudonyme :
 - a. Si vous envisagez de créer un seul pseudonyme, cliquez sur **Ajout**. Le bloc-notes Ajout d'un pseudonyme s'affiche.
 - b. Si la fonction de reconnaissance a généré une liste de pseudonymes, sélectionnez le pseudonyme pour lequel ajouter des contraintes informationnelles. Cliquez ensuite sur **Propriétés**. Le bloc-notes correspondant s'ouvre.
 - Si vous modifiez un pseudonyme, cliquez dessus. Dans le panneau des détails de l'objet du Centre de contrôle DB2, cliquez sur **Modification** dans la liste des actions. Le bloc-notes Modification du pseudonyme s'ouvre.
2. Sur la page Clés, définissez les contraintes d'intégrité référentielle pour le pseudonyme. Vous pouvez définir une contrainte de clé primaire, de clé unique ou de clé externe.
3. Sur la page Contraintes de vérification, définissez les contraintes de vérification ou de dépendance fonctionnelle pour le pseudonyme.
4. Cliquez sur **OK** pour définir les contraintes informationnelles et fermer le bloc-notes.

Indication des contraintes informationnelles sur les pseudonymes (ligne de commande DB2)

Pour améliorer les performances des requêtes sur des sources de données éloignées, vous pouvez ajouter des contraintes informationnelles aux pseudonymes. Toutefois, le serveur fédéré n'applique ou ne vérifie pas les contraintes. Vous pouvez indiquer des contraintes informationnelles sur un pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Restrictions

Après avoir défini les contraintes informationnelles sur un pseudonyme, vous pouvez modifier uniquement les noms de colonne pour ce pseudonyme après avoir supprimé les contraintes informationnelles.

A propos de cette tâche

Pour les sources de données relationnelles, vous pouvez indiquer des contraintes informationnelles lorsque vous modifiez un pseudonyme.

Pour les sources de données non relationnelles, vous pouvez indiquer des contraintes informationnelles lorsque vous créez ou modifiez un pseudonyme.

Procédure

Pour spécifier des contraintes informations sur un pseudonyme à partir de la ligne de commande DB2, émettez l'instruction CREATE NICKNAME ou ALTER NICKNAME avec les attributs de contrainte appropriés.

Spécification de contraintes informationnelles sur des pseudonymes - Exemples

Ces exemples illustrent l'utilisation des contraintes informationnelles pour les pseudonymes. Vous utilisez les instructions CREATE ou ALTER NICKNAME pour les contraintes de vérification, les contraintes référentielles et d'autres structures de données.

Exemple : Contrainte de vérification d'information

Dans la table éloignée suivante, les données de la colonne salary sont toujours supérieures à 10000.

```
CREATE TABLE account.salary (  
    empno INTEGER NOT NULL PRIMARY KEY,  
    salary INTEGER NOT NULL  
);
```

Créez un pseudonyme pour cette table :

```
CREATE NICKNAME account.salary FOR myserv.account.salary;
```

Ajoutez ensuite des informations de vérification informationnelles pour le pseudonyme en émettant l'instruction suivante :

```
ALTER NICKNAME account.salary ADD CONSTRAINT cons1 CHECK( salary > 10000 )  
NOT ENFORCED  
ENABLE QUERY OPTIMIZATION;
```

Exemple : Contrainte référentielle informationnelle : pseudonyme pour le pseudonyme

Dans cet exemple, il existe deux pseudonymes N1 et N2. La colonne F1 du pseudonyme N2 contient la valeur clé de la colonne P1 du pseudonyme N1. Vous pouvez définir la contrainte référentielle sur le pseudonyme N2 en émettant l'instruction suivante :

```
ALTER NICKNAME SCHEMA1.N2 ADD CONSTRAINT ref1  
    FOREIGN KEY (F1) REFERENCES SCHEMA1.N1 (P1)  
    NOT ENFORCED;
```

Exemple : Contrainte référentielle informationnelle : pseudonyme pour la table

Dans ce exemple, le pseudonyme N3 avec la colonne F1 contient la valeur clé dans la colonne P1 de la table T1. Vous pouvez définir la contrainte référentielle sur le pseudonyme N3 en émettant l'instruction suivante :

```
ALTER NICKNAME SCHEMA1.N3 ADD CONSTRAINT ref1  
    FOREIGN KEY (F1) REFERENCES SCHEMA1.T1 (P1)  
    NOT ENFORCED;
```

Exemple : Contrainte référentielle informationnelle : table pour le pseudonyme

Dans cet exemple, la table T2 avec la colonne F1 contient la valeur clé dans la colonne P1 du pseudonyme N4. Vous pouvez définir la contrainte référentielle sur la table T2 en émettant l'instruction suivante :

```
ALTER TABLE SCHEMA1.T2 ADD CONSTRAINT ref1
        FOREIGN KEY (F1) REFERENCES SCHEMA1.N4 (P1)
        NOT ENFORCED;
```

Exemple : Dépendance fonctionnelle

Dans cet exemple, la paire de colonnes C1 et C2 détermine uniquement la valeur dans la colonne P1. Vous pouvez définir la dépendance fonctionnelle en émettant l'instruction suivante :

```
ALTER NICKNAME SCHEMA1.NICK1 ADD CONSTRAINT FD1 CHECK( P1 DETERMINED BY (C1,C2) )
        NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Exemple : Fichier structuré en table

Cette instruction définit une clé principale pour un fichier structuré en tables :

```
CREATE NICKNAME MY_FILE (
        X INTEGER NOT NULL,
        Y INTEGER,
        PRIMARY KEY (X) NOT ENFORCED
) FOR SERVER MY_SERVER OPTIONS(FILE_PATH '/usr/pat/DRUGDATA1.TXT');
```

Schéma en étoile

L'instruction crée quatre tables de dimensions et une table de faits :

```
CREATE TABLE SCHEMA.FACT (
        LOCATION_CODE INTEGER NOT NULL,
        PRODUCT_CODE INTEGER NOT NULL,
        CUSTOMER_CODE INTEGER NOT NULL,
        SDATE DATE NOT NULL,
        SALES INTEGER NOT NULL
);

CREATE TABLE SCHEMA.LOCATION (
        LOCATION_CODE INTEGER NOT NULL PRIMARY KEY,
        STATE CHAR(2) NOT NULL,
        SHOP_ID INTEGER NOT NULL,
        ...
);

CREATE TABLE SCHEMA.PRODUCT (
        PRODUCT_CODE INTEGER NOT NULL PRIMARY KEY,
        PRODUCT_CAT INTEGER NOT NULL,
        PRODUCT_NAME VARCHAR(20) NOT NULL,
        ...
);

CREATE TABLE SCHEMA.CUSTOMER (
        CUSTOMER_CODE INTEGER NOT NULL PRIMARY KEY,
        NAME VARCHAR(20) NOT NULL,
        TEL VARCHAR(10) NOT NULL,
        ...
);

CREATE TABLE SCHEMA.TIMEDIM (
        SDATE DATE NOT NULL UNIQUE,
```

```

YEAR INTEGER NOT NULL,
QUARTER INTEGER NOT NULL,
...
);

```

Le serveur fédéré crée les pseudonymes suivants pour la table de faits et les tables de dimensions :

```

CREATE NICKNAME SCHEMA.FACT FOR SERVER.SCHEMA.FACT;
CREATE NICKNAME SCHEMA.LOCATION FOR SERVER.SCHEMA.LOCATION;
CREATE NICKNAME SCHEMA.PRODUCT FOR SERVER.SCHEMA.PRODUCT;
CREATE NICKNAME SCHEMA.CUSTOMER FOR SERVER.SCHEMA.CUSTOMER;
CREATE NICKNAME SCHEMA.TIMEDIM FOR SERVER.SCHEMA.TIMEDIM;

```

Vous pouvez définir la relation de clé externe en émettant les instructions suivantes :

```

ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT L1 FOREIGN KEY (LOCATION_CODE)
REFERENCES SCHEMA.LOCATION(LOCATION_CODE)
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT P1 FOREIGN KEY (PRODUCT_CODE)
REFERENCES SCHEMA.PRODUCT(PRODUCT_CODE)
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT C1 FOREIGN KEY (CUSTOMER_CODE)
REFERENCES SCHEMA.CUSTOMER(CUSTOMER_CODE)
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

ALTER NICKNAME SCHEMA.FACT ADD CONSTRAINT S1 FOREIGN KEY (SDATE)
REFERENCES SCHEMA.TIMEDIM(SDATE)
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Lorsque la valeur de la colonne TEL du pseudonyme CUSTOMER est unique, vous pouvez ajouter la contrainte unique informationnelle suivante avec cette instruction :

```

ALTER NICKNAME SCHEMA.CUSTOMER ADD CONSTRAINT U1 UNIQUE( TEL )
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Lorsque la valeur de la colonne SHOP_ID du pseudonyme LOCATION détermine uniquement la valeur de la colonne LOCATION_ID, vous pouvez définir la dépendance fonctionnelle suivante avec cette instruction :

```

ALTER NICKNAME SCHEMA.LOCATION
ADD CONSTRAINT F1 CHECK( LOCATION_ID DETERMINED BY SHOP_ID )
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Etant donné que la valeur de la colonne QUARTER dans le pseudonyme TIMEDIM est comprise entre 1 et 4, vous pouvez définir la contrainte de vérification informationnelle suivante avec cette instruction :

```

ALTER NICKNAME SCHEMA.TIMEDIM
ADD CONSTRAINT Q1 CHECK( QUARTER BETWEEN 1 AND 4 )
NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Les instructions de cet exemple créent des pseudonymes pour les tables éloignées. Les pseudonymes ont des clés primaires lorsque les tables éloignées ont des clés primaires. Lorsque vous créez des pseudonymes pour des vues, ils n'ont pas de clé primaire. Dans ce cas, vous pouvez changer le pseudonyme afin d'ajouter une contrainte de clé primaire informationnelle. Par exemple :

```

CREATE NICKNAME SCHEMA.LOCATION FOR SERVER.SH.V_LOCATION;
ALTER NICKNAME SCHEMA.LOCATION
ADD CONSTRAINT P1 PRIMARY KEY ( LOCATION_CODE ) NOT ENFORCED;

```

Mise à jour des statistiques de pseudonyme

L'extraction de statistiques pour un pseudonyme garantit que l'optimiseur de requêtes utilise les informations les plus récentes sur ce pseudonyme au moment de générer les plans d'accès aux requêtes.

Fonction de mise à jour des statistiques de pseudonymes - présentation

Extrayez les statistiques pour un pseudonyme afin de fournir à l'optimiseur de requêtes des informations détaillées et à jour sur le pseudonyme. L'optimiseur utilise ces informations afin de déterminer un plan d'accès optimal pour une requête.

Lorsque vous enregistrez un pseudonyme pour un objet de source de données, le serveur fédéré ajoute des informations sur l'objet de source de données au catalogue système se trouvant sur la base de données fédérée. L'optimiseur de requêtes utilise ces informations pour planifier comment extraire les données de l'objet de source de données. La base de données fédérée ne détecte pas automatiquement les modifications apportées aux objets de source de données. Les informations se trouvant dans le catalogue système peuvent donc être obsolètes.

Vous pouvez extraire les statistiques actuellement disponibles concernant les index, les colonnes et les pseudonymes de base de données à partir de la source de données éloignée. Vous pouvez extraire et mettre à jour des statistiques de pseudonyme à l'aide du Centre de contrôle DB2 ou du processeur de ligne de commande DB2.

En général, vous pouvez utiliser le Centre de contrôle DB2 ou le processeur de ligne de commande DB2 pour mettre manuellement à jour les statistiques de pseudonyme si de nouvelles statistiques sont collectées sur une table éloignée. Par ailleurs, la collecte automatique de statistiques conserve par défaut les statistiques de pseudonyme à jour.

En général, vous pouvez mettre manuellement à jour les statistiques de pseudonyme à l'aide du Centre de contrôle DB2 ou du processeur de ligne de commande DB2, sachant que la table éloignée reçoit de nouvelles statistiques. Par ailleurs, la collecte automatique de statistiques conserve par défaut les statistiques de pseudonyme à jour.

Vous pouvez extraire des statistiques de pseudonyme pour les sources de données relationnelles suivantes :

- Famille DB2 (DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Vous pouvez extraire des statistiques de pseudonyme pour les sources de données non relationnelles suivantes :

- BioRS
- Excel
- Fichiers structurés sous forme de table
- XML sur le pseudonyme racine

Les statistiques collectées pour chaque source de données varient.

Vous pouvez mettre à jour les statistiques suivantes à l'aide de la méthode utilisant les catalogues pour la collection des statistiques :

- CARD
- FPAGES
- NPAGES
- OVERFLOW
- COLCARD
- HIGH2KEY
- LOW2KEY
- NLEAF
- NLEVELS
- CLUSTERFACTOR
- CLUSTERRATIO
- FULLKEYCARD
- FIRSTKEYCARD

Vous pouvez mettre à jour les statistiques suivantes à l'aide de la méthode utilisant les données pour la collection des statistiques :

- CARD
- COLCARD
- HIGH2KEY
- LOW2KEY
- FULLKEYCARD
- FIRSTKEYCARD

Vous pouvez extraire les statistiques d'un seul pseudonyme ou de tous les pseudonymes d'un schéma sur une définition de serveur spécifique. Vous pouvez également extraire des statistiques pour les pseudonymes sur les objets de source de données avec Oracle Label Security. Si la base de données est restreinte, seuls les utilisateurs disposant des droits d'accès appropriés peuvent afficher les statistiques HIGH2KEY et LOW2KEY. Pour les bases de données non restreintes, les statistiques de pseudonyme sur les objets avec Oracle Label Security sont exposées et peuvent générer un risque de sécurité. Dans ces instances, vous pouvez restreindre l'accès aux catalogues système contenant des informations confidentielles. Si une étape de l'extraction échoue, la base de données annule les modifications.

Vous pouvez extraire les statistiques de pseudonyme sur tout système d'exploitation qui prend en charge le serveur fédéré.

Méthodes d'extraction des statistiques de pseudonyme

Vous pouvez choisir la méthode à utiliser pour la collection des statistiques et vous pouvez limiter vos choix aux colonnes et index spécifiques. La méthode de type catalogue a de meilleures performances. Par contraste, la méthode de type données fournit des statistiques plus à récentes mais son exécution est plus longue.

Vous pouvez choisir une des méthodes suivantes pour la collection des statistiques :

- Méthode utilisant les catalogues

La méthode utilisant les catalogues copie les statistiques des tables de catalogue de la source de données dans la table du catalogue fédéré. Seules les statistiques pouvant être sémantiquement mappées vers des statistiques fédérées sont copiées. Toutefois, les statistiques des pseudonymes sont aussi précises et à jour que les informations se trouvant dans le catalogue sur la source de données. Si les informations des statistiques sont obsolètes, les statistiques de pseudonyme collectées sont également obsolètes. Lorsque vous utilisez la méthode utilisant les catalogues, vérifiez que les statistiques sur la source éloignée sont à jour.

Etant donné que les statistiques sont copiées à partir du catalogue source éloigné dans le catalogue sur le serveur fédéré, la méthode utilisant les catalogues de la collection des statistiques est généralement très rapide.

- Méthode utilisant les données

La méthode utilisant les données ne dépend pas des statistiques sur la source éloignée. Cette méthode génère ses propres statistiques de manière empirique via les résultats des requêtes émis pour les pseudonymes. Avec cette méthode, les statistiques collectées reflètent de manière précise les données éloignées.

La méthode utilisant les données peut être lente si la taille des lignes des pseudonymes impliqués est importante. Les requêtes impliquent généralement des tris et des agrégats de grande taille. C'est pourquoi, choisissez une collection de statistiques utilisant les données uniquement si des statistiques satisfaisantes ne peuvent pas être obtenues à partir de la méthode utilisant les catalogues.

Si vous souhaitez augmenter les performances de la méthode utilisant les données aux dépens de la qualité des statistiques rassemblées, limitez la collection des statistiques aux types de colonne et d'index pour lesquels l'avantage est le plus important. Ces types de colonne incluent les colonnes qui sont impliquées dans les prédicats, les clés de jointure, les opérations de regroupement ou les colonnes qui font partie d'un ou de plusieurs index.

Avec la méthode utilisant les catalogues, il n'est généralement pas nécessaire de limiter la collection des statistiques à des colonnes ou à des index spécifiques, car le temps système de cette méthode de collection est très lent.

Extraction de statistiques de pseudonyme

Vous pouvez extraire les statistiques d'un pseudonyme afin de vérifier que l'optimiseur de requêtes utilise les informations de pseudonyme disponibles sur la source de données. Vous pouvez mettre à jour les statistiques de pseudonyme pour un seul pseudonyme, plusieurs pseudonymes ou tous les pseudonymes.

A propos de cette tâche

L'optimiseur de requêtes rassemble des statistiques, dont celles des pseudonymes HIGH2KEY et LOW2KEY, pour les objets de source de données utilisant le contrôle d'accès basé sur les labels ou Oracle Label Security. Pour les bases de données

utilisant ces types de sécurité, seuls les utilisateurs disposant du niveau d'accès approprié peuvent voir les statistiques. Pour les bases de données non restreintes, vous pouvez restreindre l'accès ou définir les statistiques de pseudonyme HIGH2KEY et LOW2KEY à une valeur vide ou sans signification.

Avant de commencer

Les exigences suivantes s'appliquent lorsque vous utilisez l'invite de ligne de commande pour mettre à jour les statistiques :

- Le serveur fédéré crée le fichier journal sur le serveur. Les répertoires indiqués dans le chemin doivent exister.
- Les privilèges de l'ID utilisateur isolé de l'instance fédérée doivent inclure les privilèges afin de créer le fichier journal à l'emplacement indiqué.

Restrictions

L'ID utilisateur employé pour la connexion à la base de données fédérée doit être mappé vers la table de source de données éloignée.

Si le nom ou le type de colonne local ne représente pas le mappage de type par défaut à partir du type ou du nom de colonne distant, l'utilitaire de mise à jour des statistiques de pseudonyme n'extraira pas de statistiques de colonne.

A propos de cette tâche

Par défaut, une collection de statistiques est tentée pour toutes les colonnes et tous les index d'un pseudonyme. Vous pouvez limiter les statistiques à des colonnes et index spécifiques et spécifier un fichier journal.

Procédure

Pour mettre à jour les statistiques de pseudonyme à partir des lignes de commande DB2 ou de votre application, appelez la procédure mémorisée SYSPROC.NNSTAT.

Pour mettre à jour les statistiques de pseudonyme à partir du Centre de contrôle DB2 :

1. Sélectionnez les pseudonymes pour lesquels extraire les statistiques en cours.
2. S'il n'existe pas de catalogue d'outils DB2, une fenêtre s'affiche dans laquelle vous pouvez créer le catalogue d'outils DB2. Créez le catalogue d'outils lorsque vous souhaitez planifier la mise à jour des statistiques de pseudonyme.
3. Spécifiez les paramètres de la mise à jour.

Extraction des statistiques pour plusieurs pseudonymes (Centre de contrôle DB2)

Vous pouvez mettre à jour les statistiques pour plusieurs pseudonymes à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Procédure

Pour mettre à jour des statistiques de pseudonyme pour plusieurs pseudonymes à partir du Centre de contrôle DB2 :

1. Sélectionnez les pseudonymes requis :
 - Pseudonyme avec la définition de serveur :
 - a. Développez le dossier **Objets de base de données fédérée**. Sélectionnez le dossier de l'encapsuleur que vous souhaitez utiliser.
 - b. Développez le dossier **Définitions de serveur**. Sélectionnez le dossier du serveur qui contient les pseudonymes à utiliser.
 - c. Cliquez deux fois sur le dossier **Pseudonymes**.
 - d. Cliquez avec le bouton droit sur les noms des pseudonymes et sélectionnez **Statistiques**.
 - e. Sélectionnez **Mise à jour**. Le bloc-notes Mise à jour des statistiques pour plusieurs pseudonymes s'ouvre.
 - Pour mettre à jour les statistiques pour plusieurs pseudonymes associés à une définition de base de données spécifique :
 - a. Développez le dossier **Bases de données**. Sélectionnez le dossier **Base de données** qui contient les définitions de pseudonymes à utiliser.
 - b. Cliquez deux fois sur le dossier **Pseudonymes**.
 - c. Cliquez à l'aide du bouton droit de la souris sur les noms des pseudonymes à mettre à jour et sélectionnez **Statistiques**.
 - d. Sélectionnez **Mise à jour**. Le bloc-notes Mise à jour des statistiques pour plusieurs pseudonymes s'ouvre.
2. Définissez les paramètres pour la mise à jour :
 - Page Mise à jour des statistiques :
 - a. Affichez les pseudonymes dans la fenêtre Mise à jour des statistiques et vérifiez qu'il s'agit des pseudonymes pour lesquels vous souhaitez mettre à jour les statistiques. Vous pouvez désélectionner les pseudonymes que vous ne souhaitez pas mettre à jour.
 - b. Sélectionnez **Détails** pour choisir les statistiques au niveau des colonnes et des index à mettre à jour. Le bloc-notes Détails s'affiche.
 - Page Détails :
 - a. Sélectionnez toutes les colonnes du pseudonyme à mettre à jour ou sélectionnez des colonnes spécifiques.
 - b. Sélectionnez tous les index du pseudonyme à mettre à jour ou sélectionnez des index spécifiques.
 - c. Sélectionnez un fichier journal existant ou entrez le chemin complet d'un nouveau fichier journal.
 - Page Méthode - Choisissez une des méthodes suivantes pour la collection des statistiques :
 - a. Méthode utilisant les catalogues, valide pour les pseudonymes relationnels
 - b. Méthode utilisant les données, valide pour les pseudonymes relationnels et non relationnels
 - c. Les deux méthodes, valeur par défaut
 - **Facultatif** : Sur la page Calendrier, indiquez le moment d'exécution de la mise à jour des statistiques de pseudonyme.

Extraction des statistiques pour un seul pseudonyme (Centre de contrôle DB2)

Vous pouvez extraire des statistiques pour un seul pseudonyme associé à une définition de serveur. Vous pouvez mettre à jour des statistiques de pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Procédure

Pour mettre à jour les statistiques de pseudonyme pour un pseudonyme unique à partir du Centre de contrôle DB2 :

1. Sélectionnez les pseudonymes requis.
 - Pseudonymes avec définitions de serveur :
 - a. Développez le dossier **Objets de base de données fédérée**. Sélectionnez le dossier de l'encapsuleur que vous souhaitez utiliser.
 - b. Développez le dossier **Définitions de serveur**. Sélectionnez le dossier du serveur qui contient le pseudonyme que vous souhaitez utiliser.
 - c. Cliquez deux fois sur le dossier **Pseudonymes**.
 - d. Cliquez à l'aide du bouton droit de la souris sur le pseudonyme et sélectionnez **Statistiques**.
 - e. Sélectionnez **Mise à jour**. Le bloc-notes Mise à jour des statistiques d'un seul pseudonyme s'affiche.
 - Pseudonymes avec la définition de base de données :
 - a. Développez le dossier **Bases de données**. Sélectionnez le dossier **Base de données** qui contient les définitions de pseudonymes à utiliser.
 - b. Cliquez deux fois sur le dossier **Pseudonymes**.
 - c. Cliquez à l'aide du bouton droit de la souris sur le nom du pseudonyme à mettre à jour et sélectionnez **Statistiques**.
 - d. Sélectionnez **Mise à jour**. Le bloc-notes Mise à jour des statistiques d'un seul pseudonyme s'affiche.
2. Définissez les paramètres pour la mise à jour :
 - Page Mise à jour des statistiques :
 - a. Sélectionnez toutes les colonnes du pseudonyme à mettre à jour ou sélectionnez des colonnes spécifiques.
 - b. Sélectionnez tous les index du pseudonyme à mettre à jour ou sélectionnez des index spécifiques.
 - c. Sélectionnez un fichier journal existant ou entrez le chemin complet d'un nouveau fichier journal.
 - Page Méthode - Choisissez une des méthodes suivantes pour la collection des statistiques :
 - a. Méthode utilisant les catalogues, valide pour les pseudonymes relationnels.
 - b. Méthode utilisant les données, valide pour les pseudonymes relationnels et non relationnels.
 - c. Les deux méthodes, valeur par défaut.
 - **Facultatif** : Sur la page Calendrier, indiquez le moment d'exécution de la mise à jour des statistiques de pseudonyme.

Extraction de statistiques de pseudonyme à partir de la ligne de commande - Exemples

Ces exemples indiquent comment extraire des statistiques de pseudonyme à partir de la ligne de commande en utilisant la procédure mémorisée SYSPROC.NNSTAT.

Exemple : Extraction de toutes les statistiques

Le serveur fédéré extrait les statistiques pour les pseudonymes sur le serveur DB2SERV et ne crée pas de journal.

```
CALL SYSPROC.NNSTAT('DB2SERV',NULL,NULL,NULL,NULL,0,NULL,?)
```

Extraction de toutes les statistiques pour un schéma et renvoi d'un journal

Exemple : Extraction de statistiques avec la méthode utilisant les catalogues

Le serveur fédéré extrait les statistiques pour le pseudonyme STAFF dans le schéma ADMIN. Les statistiques sont rassemblées pour les colonnes 1 à 5 et les index 1 et 2. La méthode utilisant les catalogues est utilisée pour rassembler des statistiques. Le serveur fédéré place le journal dans le fichier /home/iiuser/reportlogs/log1.txt.

```
CALL SYSPROC.NNSTAT(
  NULL, 'ADMIN','STAFF','COL1, COL2, COL3, COL4, COL5','IND1, IND2',1,
  '/home/iiuser/reportlogs/log1.txt',?)
```

Dans cet exemple, le serveur fédéré extrait les statistiques de tous les pseudonymes sur le serveur DB2Serv dans le schéma admin. Le serveur fédéré place le journal dans le fichier /home/iiuser/stats/recent.log.

```
CALL SYSPROC.NNSTAT(
  'DB2Serv', 'admin', NULL, NULL, NULL, 0, '/home/iiuser/stats/recent.log', ?)
```

Restrictions relatives aux statistiques HIGH2KEY et LOW2KEY

Lorsque vous collectez les statistiques HIGH2KEY et LOW2KEY pour un pseudonyme DB2, les informations de certaines colonnes ne sont pas collectées.

Restrictions

Lorsque l'encapsuleur DRDA crée un pseudonyme sur une table distante DB2 ou un pseudonyme, il ne collecte les statistiques HIGH2KEY et LOW2KEY que pour les colonnes numériques, et uniquement lorsque la cardinalité de la colonne est supérieure à 3.

Procédure

Pour collecter les statistiques HIGH2KEY et LOW2KEY, employez l'une des méthodes suivantes :

- Utilisez SYSPROC.NNSTAT en définissant le paramètre METHOD sur 2. Ce paramètre spécifie la collection des statistiques utilisant les données. La méthode utilisant les données interroge les données de la table éloignée afin de calculer les valeurs des statistiques locales. Cependant, cette méthode peut consommer une quantité importante de ressources sur le serveur distant et le serveur fédéré.
- Emettez l'instruction SQL UPDATE pour mettre les colonnes HIGH2KEY et LOW2KEY à jour dans la vue SYSSTAT.COLUMNS. Dans ce cas, vous devez déterminer manuellement les valeurs appropriées pour HIGH2KEY et LOW2KEY.

Création d'un catalogue d'outils DB2

Lorsque vous mettez à jour les statistiques d'un pseudonyme, vous pouvez utiliser un catalogue d'outils DB2 pour planifier des mises à jour ultérieures des statistiques de pseudonyme. Si vous ne disposez pas d'un catalogue d'outils DB2, vous êtes invité à en créer un. Vous pouvez créer un catalogue d'outils DB2 à partir du Centre de contrôle DB2 ou de l'invite de ligne de commande mais pouvez uniquement planifier la mise à jour avec le Centre de contrôle DB2.

Avant de commencer

Le serveur d'administration DB2 doit être installé.

Procédure

Pour créer un catalogue d'outils DB2 à partir du Centre de contrôle DB2 :

1. La fenêtre Mise à jour des statistiques s'affiche lorsque vous mettez à jour les statistiques de pseudonyme.
2. Sélectionnez le système sur lequel créer une base de données pour le catalogue d'outils DB2.

La base de données doit être un système catalogué n'ayant aucun stockage de métadonnées. Si le système souhaité n'est pas catalogué, vous devez effectuer cette action avant de créer la base de données pour le catalogue d'outils DB2.

Affichage de l'état des mises à niveau des statistiques de pseudonyme (Centre de contrôle DB2)

Après avoir demandé une mise à jour des statistiques pour un pseudonyme, vous pouvez afficher le statut de la mise à jour. Vous pouvez afficher le statut des mises à jour des statistiques de pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Procédure

Pour afficher le statut des mises à jour apportées aux statistiques de pseudonyme à partir du Centre de contrôle DB2 :

1. Sélectionnez **Mise à jour des statistiques**.
2. Sélectionnez **Affichage des résultats** et consultez les informations de statut.

Affichage de l'état des mises à niveau des statistiques de pseudonyme (ligne de commande DB2)

Après avoir demandé une mise à jour des statistiques pour un pseudonyme, vous pouvez afficher le statut de la mise à jour. Vous pouvez afficher le statut des mises à jour des statistiques de pseudonyme à partir du Centre de contrôle DB2 ou de la ligne de commande DB2.

Procédure

Pour afficher le statut des mises à jour apportées aux statistiques de pseudonyme à partir de la ligne de commande DB2, consultez la table SYSPROC.FED_STATS.

L'exemple suivant affiche une description de table : SYSPROC.FED_STATS (La longueur réelle des colonnes est réduite afin de simplifier l'exemple) :

```
db2 describe table sysproc.fed_stats
```

```
Column Type Type  
name schema name Length Scale Nulls
```

```
-----  
SERVER SYSIBM VARCHAR 128 0 Yes  
SCHEMA SYSIBM VARCHAR 128 0 Yes  
NICKNAME SYSIBM VARCHAR 128 0 Yes  
STATS_UPDATE_TIME SYSIBM TIMESTAMP 10 0 No  
LOG_FILE_PATH SYSIBM VARCHAR 1000 0 Yes  
SQLCODE SYSIBM INTEGER 4 0 Yes  
SQLSTATE SYSIBM CHARACTER 5 0 Yes  
STATUS SYSIBM VARCHAR 1000 0 Yes  
8 record(s) selected.
```

```
db2 "select * from sysproc.fed_stats"
```

```
SERVER SCHEMA NICKNAME STATS_UPDATE_TIME LOG_FILE_PATH SQLCODE  
-----  
ORA8 HAROLDL NICK1 2006-05-02-12.03.24.927112 - 1791 42704
```

```
SQLSTATE STATUS  
-----
```

```
SQL1791N La définition de serveur, le schéma ou le pseudonyme spécifié n'existe pas.
```

```
1 enregistrement sélectionné.
```

Procédure mémorisée SYSPROC.NNSTAT

Extraction des statistiques actuellement disponibles sur un ou plusieurs pseudonymes. Les statistiques sont sauvegardées dans le catalogue système sur la base de données fédérée.

Autorisation

SYSPROC.NNSTAT est une procédure isolée. Les privilèges de l'ID utilisateur isolé de l'instance fédérée doivent inclure les privilèges afin de créer le fichier journal à l'emplacement indiqué.

Syntaxe

```
CALL SYSPROC.NNSTAT(  
    SERVER VARCHAR(128)  
    SCHEMA VARCHAR(128)  
    NICKNAME VARCHAR(128)  
    COLNAMES CLOB(2M)  
    INDEXNAMES CLOB(2M)  
    METHOD SMALLINT  
    LOG_FILE_PATH VARCHAR(1000)  
    OUT_SQLCODE INTEGER  
    OUT_TRACE VARCHAR(2000)  
)
```

Paramètres

server Serveur sur lequel le serveur fédéré rassemble les statistiques de pseudonyme. Ce serveur correspond aux emplacements enregistrés par l'utilisateur pour définir une source de données dans la base de données fédérée. Si vous indiquez un pseudonyme, vous pouvez indiquer NULL pour ce paramètre.

Schéma

Si NULL est indiqué, le serveur fédéré extrait tous les pseudonymes sous

le serveur indiqué. Si le paramètre du serveur est NULL, le serveur fédéré extrait les statistiques du pseudonyme sous le schéma indiqué. Si le paramètre du schéma et du pseudonyme (de type nickname) sont NULL et que vous indiquez un serveur, le serveur fédéré extrait des statistiques sur le serveur indiqué.

Pseudonyme

Nom du pseudonyme. Si vous indiquez un pseudonyme, vous devez également indiquer un schéma.

Noms de colonne

Les noms des colonnes indiqués sont des identifiants de nom de colonne.

Vous pouvez indiquer ce paramètre pour un seul pseudonyme uniquement. Si vous indiquez des noms de colonne, vous devez également indiquer un schéma et un pseudonyme.

Si NULL est indiqué, les statistiques sont indiquées pour toutes les colonnes. NULL est la valeur par défaut.

Seules les colonnes indiquées sont traitées. Si une chaîne vide (") est indiquée, les colonnes ne sont pas traitées.

Noms d'index

Les noms des index indiqués sont des identifiants de nom d'index.

Vous pouvez indiquer ce paramètre pour un seul pseudonyme uniquement. Si vous indiquez des noms d'index, vous devez également indiquer un schéma et un pseudonyme. Seuls les index indiqués sont traités.

Si NULL est indiqué, les statistiques sont indiquées pour tous les index. NULL est la valeur par défaut.

Si une chaîne vide (") est indiquée, les index ne sont pas traités.

Méthode

Méthode de collecte des informations de statistique à partir de la source de données.

0 ou NULL

La méthode utilisant les catalogues est utilisée en premier. Si cette méthode n'aboutit pas, la méthode utilisant les données est utilisée. Il s'agit du paramètre par défaut.

1 Collection des statistiques utilisant les catalogues. La méthode utilisant les catalogues mappe les informations des catalogues éloignés vers les statistiques locales pour le pseudonyme. Cette méthode est valide pour les pseudonymes relationnels uniquement.

2 Collection des statistiques utilisant les données. La méthode utilisant les données interroge les données de la table éloignée afin de calculer les valeurs des statistiques locales. Cette méthode est valide pour les pseudonymes relationnels et non relationnels.

Cette méthode est la valeur par défaut des pseudonymes relationnels si la méthode utilisant les catalogues n'aboutit pas pour un pseudonyme donné. Généralement, les statistiques ne peuvent pas être collectées car les pseudonymes sont définis pour les vues éloignées. Dans ce cas, les statistiques ne sont pas disponibles sur la source éloignée.

Chemin_Fichier_Journal

Chemin et nom du fichier journal. Le serveur fédéré crée le fichier journal

sur le serveur. Les répertoires indiqués dans le chemin doivent exister. Sous Windows, utilisez deux barres obliques inverses pour indiquer le chemin du journal. Par exemple : c:\\temp\\nnstat.log. Si vous indiquez NULL, le serveur fédéré ne crée pas de journal.

Paramètres de sortie

out_SQLCode

Erreur SQL en tant que résultat des statistiques.

out_Trace

Trace.

Exemple 1 : Dans cet exemple, le serveur fédéré extrait les statistiques du pseudonyme STAFF dans le schéma ADMIN. Les statistiques sont rassemblées pour les colonnes 1, 3, 4, 6, 7 et 10 et les index 1 à 3. La méthode utilisant les données permet de collecter des statistiques. Le serveur fédéré place le journal dans le fichier /home/iiuser/reportlogs/log1.txt.

```
CALL SYSPROC.NNSTAT(  
  NULL, 'ADMIN', 'STAFF', 'COL1, COL3, COL4, COL6, COL7, COL10',  
  'IND1, IND2, IND3', 2, '/home/iiuser/reportlogs/log1.txt', ?)
```

Exemple 2 : Dans ce exemple, le serveur fédéré extrait les statistiques pour tous les pseudonymes sur le serveur DB2SERV du schéma ADMIN. Le serveur fédéré place le journal dans le fichier c:\\reports\\log1.txt.

```
CALL SYSPROC.NNSTAT(  
  'DB2SERV', 'ADMIN', NULL, NULL, NULL, 0, 'c:\\reports\\log1.txt', ?)
```

Exemple 3 : Dans cet exemple, le serveur fédéré extrait les statistiques de tous les pseudonymes sur le serveur DB2SERV et ne crée pas de journal.

```
CALL SYSPROC.NNSTAT(  
  'DB2SERV', NULL, NULL, NULL, NULL, 0, NULL, ?)
```

Mise à jour automatique des statistiques de pseudonymes

La collecte automatique de statistiques est une fonction collectant les statistiques à jour de pseudonyme et de table. Elle est par défaut activée.

Avant de commencer

Pour que la collecte automatique de statistiques s'exécute par rapport à une source de données, vérifiez qu'un mappage utilisateur est défini pour le propriétaire de l'instance, afin d'établir une connexion à cette source de données.

A propos de cette tâche

La collecte automatique de statistiques est intégrée à la fonction de maintenance automatisée des tables qui détermine à quel moment une mise à jour des statistiques de base de données est nécessaire. Pour les pseudonymes, la collecte automatique de statistiques emploie la méthode utilisant les catalogues de la procédure mémorisée (NNSTAT) des statistiques de pseudonyme. Cette méthode extrait des statistiques de pseudonyme des informations des catalogues sur le site éloigné.

Vous pouvez personnaliser la table et les pseudonymes sont pris en compte par la collecte automatique de statistiques. Par exemple, vous pouvez personnaliser les

tables traitées par la fonction de collecte automatique de statistiques et inclure ou exclure de façon spécifique tous les pseudonymes ou seulement ceux sur des serveurs déterminés.

La collecte automatique de statistiques est contrôlée par une stratégie de maintenance. Cette dernière sert aussi pour les opérations RUNSTATS automatiques et pour les statistiques de pseudonyme. Vous disposez de fichiers et d'exemples à personnaliser pour créer ou mettre à jour une stratégie.

Pour empêcher la collecte automatique des statistiques de pseudonyme, vous pouvez désactiver cette fonction ou indiquer une stratégie remplaçant celle par défaut.

Procédure

pour changer le comportement par défaut de la collecte automatique de statistiques pour la table et les pseudonymes :

- Créez une stratégie pour les opérations RUNSTATS automatiques de table. Utilisez les procédures mémorisées système SYSPROC.AUTOMAINT_SET_POLICY et SYSPROC.AUTOMAINT_SET_POLICYFILE.
- Collectez des informations de stratégie de maintenance automatique sur les opérations RUNSTATS automatiques de table. Utilisez les procédures mémorisées système SYSPROC.AUTOMAINT_GET_POLICY et SYSPROC.AUTOMAINT_GET_POLICYFILE.
- Définissez la valeur des paramètres de configuration AUTO_MAINT, AUTO_TBL_MAINT et AUTO_RUNSTATS à OFF pour désactiver la collecte automatique de statistiques.

Chapitre 13. Utilisation des données XML distantes

La fédération prend en charge le type de données XML distantes qui vous permet d'accéder et de manipuler des données XML dans une base de données DB2 Database pour Linux, UNIX et Windows.

Le système fédéré obéit à la même sémantique XML que le système de base de données DB2. Le magasin de données XML natif vous permet de stocker et d'accéder aux documents XML stockés sous forme hiérarchique en tant que colonne d'une table relationnelle. Définissez la colonne XML avec le type de données XML.

Vous pouvez créer un alias relationnel pour une table distante ou afficher le contenu du type de données XML. Vous pouvez également utiliser l'encapsuleur XML pour créer un alias non relationnel pour un document XML.

Vous pouvez ensuite utiliser l'alias dans les langages XQuery et SQL. Le langage XQuery correspond au mécanisme primaire pour la requête de documents XML. Vous pouvez utiliser le langage SQL pour effectuer des opérations de base, comme la sélection de colonnes XML et l'insertion, la mise à jour ou la suppression de données XML. Vous pouvez également intégrer les langages SQL et XQuery afin de créer des requêtes pour des données relationnelles existantes et des données XML en utilisant les fonctions SQL/XML, ainsi que les fonctions XQuery et de prédicat.

Exemples de création d'un alias pour les données XML distantes

Pour utiliser des données XML distantes, vous devez créer un alias pour une table distante contenant des données XML ou pour un document XML.

Exemple : Créez un alias relationnel correspondant à une table DB2 distante nommée XMLTABLE1. La table XMLTABLE1 contient une colonne XMLCOL qui est définie avec le type de données XML :

```
CREATE NICKNAME NNXML1 FOR SERVER1.SCHEMA1.XMLTABLE1;
```

Exemple : Créez un alias non relationnel pour un document XML à l'aide de l'encapsuleur XML :

```
CREATE NICKNAME NNXML2
  (file_path VARCHAR(64) OPTIONS(DOCUMENT 'FILE'),
  doc XML)
FOR SERVER XML_SERVER;
```

Exemples de manipulation des données XML

Après avoir créé un alias, vous pouvez demander et manipuler des données XML dans la table distante ou un document XML dans le système fédéré. Ce système prend en charge toutes les opérations de manipulation des données XML prises en charge par le système de base de données DB2.

Les exemples suivants illustrent les différentes méthodes que vous pouvez appliquer pour utiliser des données XML avec l'alias NNXML1.

Manipulation des données XML avec la fonction SQL

La fonction SQL vous permet d'effectuer des opérations de sélection, d'insertion, de mise à jour et de suppression de base.

Exemple : Utilisez les instructions SELECT et INSERT pour accéder aux données XML et insérer des données XML à partir de l'alias NNXML1.

Dans l'instruction SELECT suivante, le document XML est sélectionné dans le serveur fédéré :

```
SELECT xmlcol FROM NNXML1;
```

Dans l'instruction INSERT suivante, une chaîne est insérée dans la colonne XML d'un alias :

```
INSERT INTO NNXML1 (xmlcol) VALUES ('<a><b>My data</b></a>');
```

Manipulation des données XML à l'aide des fonctions SQL/XML

Grâce aux fonctions SQL/XML, vous pouvez effectuer tout un éventail d'opérations incluant la requête, la validation et la publication de données XML.

Exemple : Utilisez la fonction XMLVALIDATE pour valider des données XML à l'aide du schéma XML obtenu dans la spécification du schéma figurant dans le document d'instance XML :

```
SELECT XMLVALIDATE(xmlcol) FROM NNXML1;
```

Manipulation des données XML à l'aide de la fonction XQuery

Vous pouvez utiliser les fonctions XQuery pour manipuler des données XML, comme la fonction xmlcolumn.

Exemple : Utilisez la fonction XQuery pour extraire l'élément b, qui est un enfant de la racine a, de la colonne XML nommée XMLCOL :

```
xquery db2-fn:xmlcolumn('NNXML1.XMLCOL')/a/b;
```

Validation de documents XML à l'aide de schémas XML

Vous pouvez vérifier la validité d'un document XML. Pour valider le langage XML, vous devez enregistrer le schéma XML et appeler explicitement la fonction XMLVALIDATE. La validation est facultative, mais recommandée.

Enregistrement des schémas XML

L'enregistrement d'un schéma est obligatoire pour une validation XML. Vous devez également enregistrer le schéma avant de procéder à la décomposition du schéma annoté.

A propos de cette tâche

Enregistrez un schéma XML dans le référentiel de schémas XML (XSR). Le processus d'enregistrement crée un objet XSR.

L'enregistrement du schéma XML s'effectue sur le serveur fédéré.

Procédure

Pour enregistrer des objets XSR, procédez comme suit :

1. Enregistrez le document du schéma XML dans l'objet XSR en utilisant les procédures mémorisées ou des commandes via le processeur de ligne de commande.
2. Si le schéma XML contient plusieurs documents de schéma, spécifiez des documents de schéma XML supplémentaires à inclure avec l'objet XSR.
3. Terminez le processus d'enregistrement avec l'objet XSR.

Validation des documents XML

Pour valider un document XML spécifique, vous devez explicitement appeler la fonction XMLVALIDATE.

A propos de cette tâche

En règle générale, vous validez des données XML lors d'une opération d'insertion ou de mise à jour en utilisant la fonction XMLVALIDATE. Cette fonction valide une valeur XML par rapport à un schéma XML ou à celui obtenu à partir de la spécification d'un schéma dans le document d'instance.

Lorsque vous insérez ou mettez à jour une colonne XML dans un alias et que vous utilisez la fonction XMLVALIDATE, la validation s'effectue sur le serveur fédéré. Ensuite, ce dernier sérialise les données et les envoie à la source de données. Lors de la sérialisation, toutes les annotations de type de données ajoutées par XMLVALIDATE ne sont pas conservées.

Procédure

Pour valider un document XML, procédez comme suit :

Appelez la fonction XMLVALIDATE dans le cadre d'une instruction SQL.

- Si la fonction XMLVALIDATE est appelée sans aucune spécification de schéma, ce dernier est déterminé en fonction de l'attribut *schemaLocation* dans le document d'instance. S'il n'existe aucune information sur le schéma, un message d'erreur est émis.
- Si la fonction XMLVALIDATE est appelée à l'aide d'un schéma enregistré ou d'un identificateur URI (Uniform Resource Identifier), la validation s'effectue par rapport à un schéma spécifique. Si un schéma externe est spécifié, il écrase la spécification du schéma interne.

Exemple : Validez un document XML à l'aide du schéma XML MYSCHEMA.MYDOCUMENTS:

```
INSERT INTO NNXML1(XMLCOL)
VALUES (
XMLVALIDATE(
? ACCORDING TO XMLSCHEMA ID MYSCHEMA.MYDOCUMENTS
)
)
```

Si le schéma XML qui est associé à l'identificateur SQL MYSCHEMA.MYDOCUMENTS figure dans le référentiel XML, la valeur XML d'entrée est validée.

Décomposition des documents XML à l'aide des schémas XML annotés dans les alias

Grâce à la décomposition d'un schéma XML annoté, vous pouvez décomposer des documents basés sur des annotations spécifiées dans un schéma XML.

Avant de commencer

Les documents du schéma annoté doivent être stockés dans et enregistrés avec le référentiel de schémas XML (XSR).

A propos de cette tâche

Si vous avez besoin d'accéder aux données XML sous forme de données relationnelles, plutôt que dans leur forme hiérarchique, vous pouvez décomposer, ou détruire, les données XML. Vous pouvez décomposer un document XML en un alias faisant référence à une table distante.

Procédure

Pour décomposer les documents XML en un alias à l'aide des schémas XML annotés, procédez comme suit :

1. Annotez les documents du schéma en décomposant le schéma XML.
2. Créez l'alias que vous souhaitez utiliser pour la décomposition. Le nom de l'alias doit correspondre aux valeurs du schéma annoté.
3. Enregistrez le schéma à l'aide de la commande REGISTER XMLSCHEMA.
4. Attribuez les privilèges USAGE à l'objet XSR pour permettre à des utilisateurs spécifiques d'utiliser un schéma XML spécifique.
5. Activez la décomposition du schéma à l'aide de l'instruction ALTER XSROBJECT.
6. Utilisez la commande DECOMPOSE XML DOCUMENT pour décomposer le document d'instance XML.

Traitement fédéré des données XML distantes

Les données XML sont envoyées et reçues entre le système fédéré et le client de la source de données distante sous forme de chaîne XML sérialisée en format de caractères ou binaire.

Les processus affectant la façon dont le serveur fédéré envoie et reçoit les données XML sont des conventions de sérialisation, d'analyse syntaxique et de pages de codes.

Analyse syntaxique fédérée des données XML distantes

Le système fédéré utilise l'analyseur syntaxique XML DB2 pour traiter des données XML distantes.

L'analyseur syntaxique exige des données XML syntaxiquement correctes qui obéissent aux règles de codage des données. L'analyseur ne procède pas à aucune validation de schéma.

- Lors de la récupération des données, l'analyseur émet un message d'erreur si les données XML ne sont pas syntaxiquement correctes.

- Lors de l'insertion des données, selon l'emplacement où s'effectue l'analyse syntaxique, le serveur fédéré ou la source de données cible émet un message d'erreur si les données XML ne sont pas syntaxiquement correctes.

Gestion des blancs pour la fédération

Lors d'une analyse syntaxique XML, vous pouvez conserver ou supprimer un blanc de délimitation dans les documents XML. Les caractères blancs figurant entre les éléments sont des *blancs de délimitation*.

Le serveur fédéré applique les mêmes conventions pour les blancs que le système de base de données DB2.

En ce qui concerne les variables de l'hôte d'applications et les marqueurs de paramètres, le registre spécial CURRENT IMPLICIT XMLPARSE OPTION détermine si le blanc est supprimé lors de l'opération de liaison fédérée. Si la source de données respecte différentes règles pour la gestion des blancs, le système fédéré essaie de compenser la différence dans la sémantique.

Emissions des pages de codes pour les données XML distantes

Certaines émissions de pages de codes peuvent affecter des instructions fédérées.

Le serveur fédéré obéit aux règles de codage de l'analyseur syntaxique XML de DB2.

En ce qui concerne les instructions fédérées, les problèmes suivants peuvent se produire lors de la manipulation de données XML distantes sérialisées :

- Pour les données XML distantes qui sont sérialisées au format binaire :
 - Lorsque des données sont envoyées depuis le client distant à l'encapsuleur fédéré, aucune perte de données n'est occasionnée.
 - Si un codage interne existe et qu'il n'est pas un schéma de codage IBM valide, l'encapsuleur remplace l'attribut de codage par un schéma de codage IBM valide ou remplace l'attribut de codage à partir de la déclaration XML et l'analyseur syntaxique XML de DB2 décode la valeur.
- Pour les données XML distantes qui sont sérialisées au format de caractères :
 - La page de codes des données XML figure dans la page de codes de la base de données fédérée. Lorsque le client de la source de données envoie des données à l'encapsuleur fédéré, des pertes de données peuvent être occasionnées. Si la conversion génère une substitution de caractères, un avertissement peut être émis selon le comportement de la source de données et l'implémentation de l'encapsuleur.
 - S'il existe un codage interne, l'encapsuleur le supprime car la chaîne sérialisée figure dans la page de codes de la base de données fédérée.

Pour ce qui est des encapsuleurs non relationnels, l'analyseur syntaxique XML de DB2 décode les données XML à l'aide du codage interne du document.

Restrictions appliquées au type de données XML distantes

Les systèmes fédérés imposent des restrictions sur le type de données XML distantes.

Vous pouvez uniquement utiliser le type de données XML distantes avec la source de données DB2 pour Linux, UNIX et Windows et l'encapsuleur XML.

Vous ne pouvez pas effectuer les opérations SQL suivantes :

- Modifier le type de données XML d'un alias, ce qui génère une erreur SQL0270N
- Créer un index fédéré SPECIFICATION ONLY avec la clause `xml-index-specification`, ce qui génère une erreur SQL0104N
- Définir une procédure mémorisée fédérée avec le type de données XML, ce qui génère une erreur SQL1254N
- Créer un mappage type entre le type de données XML et un autre type de données, ce qui génère une erreur SQL0604N

Les restrictions suivantes s'appliquent aux colonnes XML de l'encapsuleur XML :

- Vous ne pouvez pas spécifier d'options de colonne pour une colonne XML.
- Seul l'alias racine peut contenir une seule colonne XML faisant référence à l'ensemble du contenu d'un document XML.
- Chaque alias racine doit correspondre à un document XML unique.
- Les alias enfant, les options XPath d'un alias et les options de noms d'espace d'un alias ne sont pas pertinents.

Chapitre 14. Tolérances des erreurs dans les expressions de tables imbriquées

La tolérance aux erreurs est un mécanisme qui permet la poursuite de l'exécution des requêtes tout en tolérant certaines erreurs SQL dans des expressions de tables imbriquées. Avec la tolérance aux erreurs, au lieu de recevoir une erreur pour une partie d'une requête et que l'intégralité de la requête s'arrête, vous pouvez obtenir un nombre maximal de résultats de requête à partir des données disponibles.

Lorsque le serveur fédéré rencontre une erreur autorisée, le serveur tolère l'erreur et continue le traitement de la requête au lieu de renvoyer une erreur pour l'intégralité de la requête. L'ensemble de résultats renvoyé par le serveur fédéré peut être un résultat vide ou partiel.

Lorsque le serveur fédéré tolère des erreurs, il renvoie les résultats de la requête même lorsque les sources de données auxquelles accède la requête ne sont pas disponibles. Ce mécanisme est utile lorsque vous souhaitez renvoyer l'ensemble des informations disponibles, malgré des résultats de requête incomplets. Par exemple, prenons l'exemple d'un médecin qui a besoin de données sur un type spécifique de condition médicale. Une requête est exécutée pour rassembler des informations à partir des sources de données de différents hôpitaux. Si une ou plusieurs bases de données d'hôpital ne sont pas disponibles, les résultats provenant uniquement des bases de données disponibles sont toujours très utiles pour le médecin.

Les requêtes qui contiennent des branches UNION ALL peuvent tirer profit de la tolérance aux erreurs. Sans ce mécanisme, si le traitement d'une branche de la requête rencontre une erreur, le serveur fédéré arrête le traitement de la requête. Avec ce mécanisme, lorsque vous indiquez l'erreur devant être tolérée sur la même branche de la requête, le serveur fédéré tolère l'erreur et continue de parcourir le reste des branches disponibles. L'opération UNION ALL renvoie les résultats provenant des sources de données disponibles.

Exemple : La requête suivante sélectionne des données parmi trois pseudonymes sur trois différents serveurs :

```
SELECT c1 from
pseudonyme1_sur_serveur1
UNION ALL
SELECT c1 from pseudonyme2_sur_serveur2
UNION ALL
SELECT c1 from pseudonyme3_sur_serveur3
```

Lorsque `pseudonyme2_sur_serveur2` n'est pas disponible ou lorsque le serveur éloigné `serveur2` n'est pas disponible lors du traitement des requêtes, vous pouvez obtenir l'ensemble de résultats à partir de l'élément `pseudonyme1_sur_serveur1` et de l'élément `pseudonyme3_sur_serveur3` en tolérant les erreurs avec `pseudonyme2` et `serveur2`. Un ensemble de résultats provenant de deux ou de trois branches de requête est équivalent à l'exécution de la requête suivante :

```
SELECT c1 from
pseudonyme1_sur_serveur1
UNION ALL
SELECT c1 from pseudonyme3_sur_serveur3
```

Vous pouvez indiquer les erreurs SQL que vous souhaitez autoriser dans une expression de table imbriquée lors du traitement des requêtes. Les types d'erreur que le serveur fédéré tolère sont des erreurs avec des connexion éloignées, l'autorisation et l'authentification.

Spécification pour les expressions de tables imbriquées pour la tolérance aux erreurs

Vous indiquez les erreurs à tolérer dans une expression de table imbriquée avec la clause RETURN DATA UNTIL.

A propos de cette tâche

Lorsque vous utilisez la clause RETURN DATA UNTIL, vous devez indiquer les codes d'erreur que vous souhaitez tolérer. Le tableau suivant répertorie les erreurs qui sont admises dans la clause *valeur-condition-spécifique*. Vous devez indiquer un élément SQLSTATE ou un élément SQLSTATE et SQLCODE qui correspond à un code d'erreur admissible. Les codes SQL répertoriés dans le tableau sont requis.

Tableau 19. Erreur admises dans la clause valeur-condition-spécifique

SQLSTATE	Code d'erreur	SQLCODE
08001	SQL30080N	-30080
08001	SQL30081N	-30081
08001	SQL30082N	-30082
08001	SQL1336N	-1336
08004	Tous	Tout code
28000	Tous	Tous
42501	Tous	Tous
42512	Tous	Tous
42704	SQL0204N	-204
42720	Tous	Tous

Procédure

Pour spécifier les expressions de tables imbriquées pour la tolérance aux erreurs, créez une instruction SQL qui contient la clause RETURN DATA UNTIL.

RETURN DATA UNTIL *valeur-condition-spécifique*

RETURN DATA UNTIL

Toutes les lignes renvoyées de l'instruction fullselect, avant la condition spécifiée, sont renvoyées dans l'ensemble de résultats à partir de fullselect.

valeur-condition-spécifique

Définit la condition et les valeurs pour la tolérance aux erreurs.

FEDERATED

Mot clé requis. La condition spécifique que vous indiquez doit inclure uniquement les erreurs qui se produisent sur une source de données fédérée.

SQLSTATE VALUE *constante-chaîne*

Vous pouvez indiquer une condition spécifique en tant que valeur SQLSTATE. La longueur de la constante de chaîne doit être 5

lorsque VALUE est indiqué. Une valeur SQLSTATE peut être restreinte à une valeur SQLCODE particulière. Vous pouvez indiquer plusieurs valeurs SQLCODE qui partagent la même valeur SQLSTATE dans une *valeur-condition-spécifique*.

Expressions de tables imbriquées pour la tolérance aux erreurs - Exemple

Les exemples suivants illustrent le mode d'utilisation de la clause RETURN DATA UNTIL pour le renvoi des résultats de la requête lorsqu'une ou plusieurs sources de données fédérées ne sont pas disponibles.

Exemple : L'instruction SQL suivante sélectionne des données dans trois différents serveurs : SQLServer, Oracle et Sybase.

```
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN(SELECT c1 FROM n1_from_SQLServer) etq1
UNION ALL
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN (SELECT c1 FROM n2_from_Oracle) etq2
UNION ALL
SELECT c1 FROM
TABLE RETURN DATA UNTIL
FEDERATED SQLSTATE '08001' SQLCODE -30080, -30082
WITHIN(SELECT c1 FROM n3_from_Sybase) etq3;
```

Scénario 1 : Un serveur n'est pas disponible.

Dans ce scénario, le serveur Oracle n'est pas disponible. Les serveurs SQLServer et Sybase sont disponibles. Dans cette situation, la requête dans la deuxième branche de l'opération UNION ALL renvoie un ensemble de résultats avec l'erreur 30080, qui est définie de telle sorte qu'elle soit tolérée. La requête renvoie les résultats à partir de n1_from_SQLServer et de n3_from_Sybase. L'avertissement sqlwarn5='E' est émis.

L'ensemble de résultats est équivalent à l'exécution de la requête suivante :

```
SELECT c1 FROM n1_from_SQLServer
UNION ALL
SELECT c1 FROM n3_from_Sybase;
```

Scénario 2 : Tous les serveurs sont indisponibles.

Dans ce scénario, tous les serveurs (SQLServer, Oracle, et Sybase) sont indisponibles. Dans ce cas, l'opération UNION ALL renvoie un ensemble de résultats vide. L'avertissement sqlwarn5='E' est émis.

Scénario 3 : Tous les serveurs sont disponibles.

Si tous les serveurs sont disponibles, l'ensemble de résultats de la requête est équivalent à l'exécution de la même requête sans la spécification de la clause RETURN DATA UNTIL.

Support de source de données pour les expressions de tables imbriquée pour la tolérance aux erreurs

La tolérance aux erreurs est prise en charge pour plusieurs sources de données relationnelles et pour les pseudonymes non relationnels.

La tolérance aux erreurs dans les expressions de tables imbriquées est prise en charge pour les sources de données relationnelles suivantes :

- Famille DB2 (DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Vous pouvez utiliser des pseudonymes non relationnels dans des expressions de tables imbriquées pour la tolérance aux erreurs. Le serveur fédéré peut tolérer les erreurs d'autorisation, d'authentification ou de connexion admissibles lorsque les encapsuleurs non relationnels renvoient un code d'erreur admissible.

Restrictions applicables aux expressions de tables imbriquées pour la tolérances aux erreurs

Certaines restrictions s'appliquent lorsque vous définissez des expressions de tables imbriquées tolérantes aux erreurs.

Une requête ou une vue est en lecture seule lorsque vous définissez la requête ou la vue avec une expression qui contient la clause RETURN DATA UNTIL. Les curseurs qui sont déclarés dans des expressions avec la clause RETURN DATA UNTIL sont en lecture seule. Des erreurs sont renvoyées pour chacune de ces situations.

Chapitre 15. Surveillance des serveurs et pseudonymes fédérés

Vous pouvez utiliser les éléments du Moniteur de santé et du Moniteur des systèmes pour surveiller un système fédéré.

Indicateurs de santé pour les pseudonymes et serveurs fédérés

Vous pouvez utiliser des indicateurs dans le Centre de santéDB2 pour surveiller l'état de vos pseudonymes et serveurs fédérés.

L'indicateur de santé des pseudonymes est `db.fed_nicknames_op_status`. L'indicateur de santé des définitions de serveur est `db.fed_servers_op_status`. Les indicateurs de santé fédérés sont installés en même temps que le Moniteur de santé.

Par défaut, le Centre de santé n'active pas les indicateurs de santé fédérés. Vous devez activer les indicateurs.

Lorsque l'état d'un pseudonyme ou d'un serveur n'est pas normal, les indicateurs de santé émettent une alerte. Vous pouvez voir les résultats de la surveillance en utilisant le Centre de santé ou la ligne de commande.

Les serveurs fédérés utilisant les systèmes d'exploitation AIX, HP-UX, Linux, Microsoft Windows et Solaris prennent en charge les indicateurs de santé.

tableau 20 décrit les indicateurs de santé pour les pseudonymes et serveurs fédérés.

Tableau 20. Indicateurs de santé de pseudonyme et de serveur

Indicateur de santé	Description
<code>db.fed_nicknames_op_status</code>	<p>Indique la santé de l'agrégat de tous les pseudonymes relationnels définis dans une base de données sur un serveur fédéré.</p> <p>Vous alerte si un pseudonyme n'est pas valide. Fournit des détails sur les pseudonymes qui ne sont pas valides et propose des actions à prendre pour les réparer.</p>
<code>db.fed_servers_op_status</code>	<p>Indique la santé de l'agrégat de tous les serveurs fédérés définis dans une base de données sur un serveur fédéré.</p> <p>Vous alerte si un serveur est indisponible. Fournit des détails sur les serveurs indisponibles et propose des actions à prendre pour les rendre disponibles.</p>

Les indicateurs de santé peuvent évaluer les sources de données suivantes :

- Famille DB2 (DRDA)
- Excel
- Informix
- JDBC

- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Fichiers structurés sous forme de table
- Teradata
- XML (uniquement pour les pseudonymes racines)

Activation des indicateurs de santé fédérés

Pour surveiller la santé des pseudonymes et serveurs, vous devez activer les indicateurs de santé fédérés. L'indicateur de santé des pseudonymes est `db.fed_nicknames_op_status`. L'indicateur de santé des définitions de serveur est `db.fed_servers_op_status`.

Procédure

Pour activer les indicateurs de santé fédérés, vous pouvez ouvrir le Centre de santé DB2 et configurer les indicateurs de santé ou utiliser l'interpréteur de commandes.

Surveillance de la santé des pseudonymes et serveurs fédérés

La surveillance de l'état des pseudonymes et serveurs peut vous aider à déterminer et à résoudre des problèmes dans votre système fédéré. Vous pouvez surveiller l'état de pseudonymes et serveurs fédérés en utilisant des indicateurs de santé dans le Centre de santé.

Avant de commencer

- Assurez-vous que les privilèges SELECT sur les pseudonymes sont définis sur le serveur fédéré.
- Affectez la valeur YES au paramètre de configuration de gestionnaire de base de données FEDERATED_PARALLEL.
- Si la source de données nécessite une authentification, l'ID du Moniteur de santé doit fournir des mappages utilisateur à la source de données. Le Moniteur de santé utilise ce mappage pour se connecter à la source de données.

Restrictions

«Indicateurs de santé pour les pseudonymes et serveurs fédérés», à la page 197 liste les sources de données que les indicateurs de santé peuvent évaluer.

A propos de cette tâche

Vous pouvez voir les résultats de la surveillance en utilisant le Centre de santé ou la ligne de commande. Utilisez le Centre de contrôle DB2 ou le processeur de ligne de commande DB2 pour résoudre les problèmes identifiés par les indicateurs de santé.

Procédure

Pour effectuer cette tâche à partir de la ligne de commande, émettez la commande `GET HEALTH SNAPSHOT`.

Pour effectuer cette tâche à partir du Centre de contrôle DB2 :

1. Ouvrez le Centre de santé.
2. Ouvrez l'assistant de recommandations pour visualiser des recommandations sur la résolution des pseudonymes non valides ou des serveurs indisponibles.
3. Pour effectuer cette tâche à partir de la ligne de commande, émettez la commande GET HEALTH SNAPSHOT.

Surveillance de la santé des pseudonymes et serveurs fédérés - exemple

Cette rubrique fournit un exemple d'instantané de santé d'une base de données.

Les noms des indicateurs de santé fédérés sont db.fed_nicknames_op_status et db.fed_servers_op_status. Vous devez activer ces indicateurs de santé en utilisant le Centre de santé ou les commandes suivantes dans l'interpréteur de commandes :

```
db2 update alert cfg for databases using db.fed_nicknames_op_status set
THRESHOLDSCHECKED YES
db2 update alert cfg for databases using db.fed_servers_op_status set
THRESHOLDSCHECKED YES
```

La commande suivante récupère un instantané de santé de base de données incluant les indicateurs de santé fédérés, s'ils ont été activés :

```
db2 get health snapshot for database on <nom_basedonnées>
```

Dans cet exemple, le nom de la base de données est fedhi. La sortie de cette commande indique que l'état des deux indicateurs de santé est normal. Normal signifie que les pseudonymes et les serveurs sont valides.

Instantané de santé de base de données

```
Horodatage de l'instantané                = 02/10/2006 12:10:55.063004
Nom de la base de données                 = FEDHI
Chemin d'accès à la base de données      = C:\DB2\NODE0000\SQL00006\
Alias de base de données d'entrée        = FEDHI
Système d'exploitation exécuté sur le serveur de la base= NT
Emplacement de la base de données       = Local
Etat d'alerte de gravité le plus élevé de la base de données = Attention
```

Indicateurs de santé :

```
Nom de l'indicateur                      = db.fed_servers_op_status
Valeur                                    = 0
Hodatage d'évaluation                     = 02/10/2006 12:09:10.961000
Etat d'alerte                             = Normal

Nom de l'indicateur                      = db.fed_nicknames_op_status
Valeur                                    = 0
Hodatage d'évaluation                     = 02/10/2006 12:09:10.961000
Etat d'alerte                             = Normal

Nom de l'indicateur                      = db.db_op_status
Valeur                                    = 0
Hodatage d'évaluation                     = 02/10/2006 12:08:10.774000
Etat d'alerte                             = Normal

Nom de l'indicateur                      = db.sort_shrmem_util
Valeur                                    = 0
Unité                                      = %
Hodatage d'évaluation                     = 02/10/2006 12:08:10.774000
Etat d'alerte                             = Normal
```

Nom de l'indicateur	= db.spilled_sorts
Valeur	= 0
Unité	= %
Hodatage d'évaluation	= 02/10/2006 12:09:10.961000
Etat d'alerte	= Normal

Surveillance d'instantané des systèmes fédérés - Présentation

Vous pouvez utiliser le moniteur d'instantané pour capturer des informations sur les sources de données fédérées et les applications connectées à un moment donné.

Les instantanés permettent de déterminer l'état d'un système fédéré. Pris à intervalles réguliers, les instantanés permettent également d'observer les tendances et de prévoir des problèmes potentiels.

La sortie du moniteur d'instantané est disponible dans les formats suivants :

- Sous forme de texte, par l'intermédiaire de l'interface de processeur de ligne de commande de moniteur d'instantané.
- Comme sortie de fonctions de table. Cette sortie permet d'écrire des requêtes qui limitent la sortie.

Les instantanés suivants sont particulièrement utiles pour les charges de travail fédérées :

Instantané d'instruction SQL dynamique

Fournit un instantané de toutes les instructions SQL dynamique dans le cache d'instruction incluant les instructions fédérées et non fédérées.

Instantané d'application

Fournit des informations sur une application particulière, incluant le texte de l'instruction SQL en cours d'exécution.

Instantané de bases de données distantes

Fournit des informations sur une base de données particulière de système fédéré.

Instantané de toutes les bases de données distantes

Fournit des informations sur chaque base de données de système fédéré active.

Instantané d'applications distantes

Fournit des informations sur chaque application de système fédéré active.

Surveillance des requêtes fédérées

En surveillant les requêtes, vous pouvez déterminer la façon dont votre système fédéré est exécuté. Pour vous aider à comprendre comment votre système fédéré traite une requête, vous pouvez obtenir un instantané de la requête distante.

A propos de cette tâche

Le moniteur d'instantané suit deux aspects de chaque requête traitée par le serveur fédéré :

- Toute la requête fédérée est soumise par l'application qui référence des pseudonymes, des tables locales ou les deux.

- Pour les requêtes impliquant des pseudonymes, un ou plusieurs *fragments distants*. Les fragments distants sont des instructions automatiquement générées et soumises aux sources de données distantes dans leur dialecte natif pour la requête fédérée.

La surveillance des requêtes fédérées exige que vous teniez compte à la fois du travail effectué localement sur le serveur fédéré et du travail effectué sur les serveurs distants en réponse aux fragments de requêtes distantes. L'instantané d'instruction SQL dynamique et la fonction de table `SNAPSHOT_DYN_SQL` contiennent des informations sur les requêtes fédérées individuelles lorsqu'elles sont soumises au serveur fédéré et sur les fragments de requête distante que le serveur fédéré envoie aux autres sources de données.

Avant de commencer

Vous devez affecter la valeur `ON` à l'inverseur logique du moniteur `STATEMENT` pour la base de données fédérée pour collecter des informations d'instantané sur les requêtes distantes.

Procédure

Pour surveiller des requêtes sur le serveur fédéré en étant connecté à la base de données fédérée, utilisez une des méthodes suivantes :

- Sortie textuelle :

```
GET SNAPSHOT FOR DYNAMIC SQL on nombd
```

où *nombd* est le nom de la base de données du serveur fédéré.

- Fonction de table :

```
CREATE TABLE table_snap AS (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL ('nombd', -1))
  as snaptab) definition only;
INSERT INTO snap (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL ('nombd', -1))
  as snaptab);
```

Vous pouvez alors écrire une requête par rapport à la table d'instantané contenant une ligne par requête (fédérée ou non fédérée) et une ligne par fragment de requête dans le cache d'instruction du serveur.

Le nom des fragments de requête distante est le serveur auquel ils ont été envoyés ajouté en préfixe au texte de la requête distante, entre crochets, dans la zone `stmt_text` de la fonction de table. Par exemple, vous pouvez utiliser la requête suivante pour rechercher des fragments distants à exécution longue :

```
SELECT total_exec_time, rows_read, total_usr_cpu_time, num_executions,
  substr(stmt_text,1,30)
FROM TABLE(SNAPSHOT_DYN_SQL ('nombd', -1))AS snaptab
-- remote fragments only
WHERE stmt_text LIKE '[%]%'
ORDER BY total_exec_time;
```

En comparant le temps d'exécution d'une instruction fédérée complète aux temps d'exécution de fragments distants envoyés à d'autres sources de données de la part de l'instruction, vous pouvez comprendre où se passe la majorité du temps de traitement.

Pour déterminer quels fragments de requête sont envoyés aux sources distantes de la part d'une requête fédérée, consultez un plan d'exécution `EXPLAIN` pour la requête.

Surveillance d'instantané des requêtes fédérées - exemple

Cette rubrique fournit un exemple de sortie pour l'instantané SQL dynamique à base de texte d'une requête fédérée impliquant une source de données Oracle distante.

L'instruction suivante récupère un instantané de toutes les instructions actuellement dans le cache d'instruction, dont les instructions fédérées et les fragments distants envoyés à d'autres sources de données :

```
GET SNAPSHOT FOR DYNAMIC SQL ON <nom_basededonnées>
```

Le nom de base de données est le nom de la base de données fédérée locale.

Dans l'exemple suivant, la sortie est le résultat de l'instruction :

```
GET SNAPSHOT FOR DYNAMIC SQL ON FEDDB
```

L'exemple montre une instruction fédérée et un fragment distant envoyé par cette instruction fédérée. Les fragments distants sont identifiés par le nom de serveur distant ajouté en préfixe au texte de l'instruction distante entre crochets. Dans cet exemple, le serveur Oracle distant est nommé ORA9. La première entrée indique l'instruction SQL fédérée référençant les pseudonymes avec la durée totale écoulée. La deuxième entrée indique l'instruction distante envoyée à la source [ORA9] référençant les noms de tables Oracle distantes.

Résultat de l'instantané SQL dynamique

```
Nom de la base de données                = FEDDB

Nombre d'exécutions                      = 1
Nombre de compilations                   = 1
Temps de préparation le plus mauvais (ms) = 475
Temps de préparation le meilleur (ms)    = 475
Lignes internes supprimées                = 0
Lignes internes insérées                 = 0
Lignes lues                              = 5
Lignes internes mises à jour              = 0
Lignes écrites                           = 0
Tris d'instructions                      = 0
Dépassements de tris d'instructions       = 0
Durée de tri totale                      = 0
Lectures logiques de données du pool de mémoire tampon = non collecté
Lectures physiques de données du pool de mémoire tampon = non collecté
Lectures logiques de données temporaires du pool de mémoire tampon = non collecté
Lectures physiques de données temporaires du pool de mémoire tampon = non collecté
Lectures logiques d'index du pool de mémoire tampon = non collecté
Lectures physiques d'index du pool de mémoire tampon = non collecté
Lectures logiques d'index temporaires du pool de mémoire tampon = non collecté
Lectures physiques d'index temporaires du pool de mémoire tampon = non collecté
Lectures logiques de XDA du pool de mémoire tampon = non collecté
Lectures physiques de XDA du pool de mémoire tampon = non collecté
Lectures logiques de XDA temporaires du pool de mémoire tampon = non collecté
Lectures physiques de XDA temporaires du pool de mémoire tampon = non collecté
Durée d'exécution totale (sec.ms)        = 1.816884
Temps UC utilisateur total (sec.ms)      = 0.000000
Temps UC système total (sec.ms)          = 0.020000
Texte de l'instruction                   = select count(*) from orat.supplier,
    orat.nation where s_nationkey =
    n_nationkey and n_name <> 'FRANCE'

Nombre d'exécutions                      = 1
Nombre de compilations                   = 1
Temps de préparation le plus mauvais (ms) = 0
```



```

Temps de préparation le meilleur (ms)          = 0
Lignes internes supprimées                    = 0
Lignes internes insérées                      = 0
Lignes lues                                  = 1
Lignes internes mises à jour                  = 0
Lignes écrites                               = 0
Tris d'instructions                          = 0
Dépassements de tris d'instructions          = 0
Durée de tri totale                          = 0
Lectures logiques de données du pool de mémoire tampon = non collecté
Lectures physiques de données du pool de mémoire tampon = non collecté
Lectures logiques de données temporaires du pool de mémoire tampon = non collecté
Lectures physiques de données temporaires du pool de mémoire tampon = non collecté
Lectures logiques d'index du pool de mémoire tampon = non collecté
Lectures physiques d'index du pool de mémoire tampon = non collecté
Lectures logiques d'index temporaires du pool de mémoire tampon = non collecté
Lectures physiques d'index temporaires du pool de mémoire tampon = non collecté
Lectures logiques de XDA du pool de mémoire tampon = non collecté
Lectures physiques de XDA du pool de mémoire tampon = non collecté
Lectures logiques de XDA temporaires du pool de mémoire tampon = non collecté
Lectures physiques de XDA temporaires du pool de mémoire tampon = non collecté
Durée d'exécution totale (sec.ms)           = 1.337672
Temps UC utilisateur total (sec.ms)         = 0.000000
Temps UC système total (sec.ms)            = 0.000000
Texte de l'instruction                      = [ORA9] SELECT COUNT(*) FROM "TPCH"."NATION"
      A0, "TPCH"."SUPPLIER" A1 WHERE
      (A0."N_NAME" <> 'FRANCE          ') AND
      (A1."S_NATIONKEY" = A0."N_NATIONKEY")

```

L'instantané n'a pas collecté d'informations de pool de mémoire tampon car celles-ci ne sont pas applicables aux requêtes distantes.

Éléments du moniteur des systèmes de base de données fédérée

Cette rubrique décrit les éléments du moniteur fournissant des informations sur les systèmes fédérés.

Un système fédéré accède à différentes sources de données qui peuvent résider sur différentes plateformes, d'IBM et d'autres fournisseurs, relationnelles et non relationnelles. Il intègre un accès aux données distribuées et présente une seule image de base de données d'un environnement hétérogène à ses utilisateurs.

Les éléments suivants donnent des informations sur l'accès total à une source de données par des applications exécutées dans un système fédéré et des informations sur l'accès à une source de données par une application donnée exécutée dans une instance de système fédéré. Ces éléments sont :

- datasource_name - élément du moniteur Nom de source de données
- disconnects - élément du moniteur Déconnexions
- insert_sql_stmts - élément du moniteur Insertions
- update_sql_stmts - élément du moniteur Mises à jour
- delete_sql_stmts - élément du moniteur Suppressions
- dynamic_sql_stmts - élément du moniteur Instructions SQL dynamique tentées
- create_nickname - élément du moniteur Créer des pseudonymes
- passthru - élément du moniteur Passe-système
- stored_procs - élément du moniteur Procédures mémorisées
- remote_locks - élément du moniteur Verrous distants
- sp_rows_selected - élément du moniteur Lignes renvoyées par les procédures mémorisées

- `select_time` - élément du moniteur Temps de réponse de requête
- `insert_time` - élément du moniteur Temps de réponse d'insertion
- `update_time` - élément du moniteur Temps de réponse de mise à jour
- `delete_time` - élément du moniteur Temps de réponse de suppression
- `create_nickname_time` - élément du moniteur Temps de réponse de création de pseudonyme
- `passthru_time` - élément du moniteur Temps de passe-système
- `stored_proc_time` - élément du moniteur Temps de procédure mémorisée
- `remote_lock_time` - élément du moniteur Temps de verrou distant

L'exemple suivant montre l'instantané `dynamic_sql_statement` :

```
Statement text = [ORACLE817]SELECT A0.C1,A0.C2 FROM ORA_T A0 WHERE A0.C3 = :H0
```

Pour toutes les instructions distantes, le texte de l'instruction commence par le nom de la source de données distante entre crochets, suivi du texte envoyé à la source de données distante.

Chapitre 16. Mode d'interaction des applications client avec les sources de données

Pour les applications client, les sources de données dans un système fédéré apparaissent comme une seule base de données collective. Pour obtenir des données des sources de données, les applications soumettent des requêtes en langage DB2 à la base de données fédérée. La base de données fédérée distribue ensuite les requêtes aux sources de données appropriées et renvoie ces données aux applications ou effectue l'action demandée.

La base de données peut joindre des données à partir de tables locales ou de sources de données éloignées dans une même instruction SQL. Par exemple, vous pouvez joindre des données se trouvant dans une table DB2 locale, dans une table Informix et dans une vue Sybase dans une même instruction SQL. En traitant des instructions SQL comme si les sources de données étaient des vues ou des tables relationnelles ordinaires dans la base de données fédérée, le système fédéré peut joindre des données relationnelles avec des données non relationnelles.

Dans un système fédéré, vous pouvez accéder aux sources de données par l'intermédiaire de pseudonymes. Un *pseudonyme* est un objet de base de données fédérée qu'une application utilise pour référencer un objet de source de données, tel qu'une table ou une vue. Pour écrire dans une source de données, pour mettre à jour une table de source de données par exemple, une application peut utiliser SQL DB2 (avec des pseudonymes). Une autre solution pour les applications est d'utiliser le dialecte SQL de la source de données (sans pseudonyme) dans une session spécifique appelée *passer-système* pour accéder directement aux sources de données.

Les applications qui utilisent SQL DB2 et les pseudonymes peuvent accéder à tous les types de données reconnus par la base de données fédérée.

Le catalogue de la base de données fédérée contient des informations relatives aux objets se trouvant dans la base de données fédérée et des informations relatives aux objets des sources de données. Dans la mesure où le catalogue contient des informations relatives à l'intégralité de la base de données fédérée, il est appelé *catalogue global*.

Chapitre 17. Référence aux objets de source de données à l'aide de pseudonymes dans des instructions SQL

Avec un système fédéré, vous utilisez les pseudonymes définis pour les objets de source de données pour représenter les objets dans vos instructions SQL. Le système fédéré ne reconnaît pas les noms qualifiés complets d'objets, de schémas et de sources de données dans les instructions SQL.

Des pseudonymes doivent être enregistrés dans la base de données fédérée pour les objets de source de données pour que vous puissiez inclure ceux-ci dans vos requêtes. En général, vous pouvez spécifier des pseudonymes dans une instruction SQL là où vous pouvez spécifier des tables locales.

Exemple : utilisation de pseudonymes dans les instructions SELECT, INSERT, UPDATE et DELETE

Vous définissez le pseudonyme NFXDEPT pour représenter une table dans une table Informix appelée PERSON.DEPT, où :

- PERSON est le schéma de source de données
- DEPT est le nom de la table de source de données

L'instruction `SELECT * FROM NFXDEPT` est autorisée à partir du serveur fédéré. En revanche, l'instruction `SELECT * FROM PERSON.DEPT` n'est pas autorisée (sauf dans une session passe-système). PERSON.DEPT n'est pas enregistré en tant que pseudonyme sur le serveur fédéré.

Exemple : utilisation de pseudonymes dans l'instruction CREATE TABLE

Vous voulez créer une table locale en fonction d'une table distante pour laquelle vous avez défini un pseudonyme. Exemple d'instruction `CREATE TABLE` :

```
CREATE TABLE nom_table LIKE pseudonyme
```

Pseudonymes dans les instructions DDL

Des pseudonymes doivent être enregistrés dans la base de données fédérée pour les objets de source de données pour que vous puissiez inclure ceux-ci dans vos instructions DDL. Cette rubrique fournit des exemples d'instructions DDL que vous pouvez utiliser avec des systèmes fédérés.

Utilisation de pseudonymes dans l'instruction COMMENT ON

L'instruction `COMMENT ON` ajoute ou remplace des commentaires dans le catalogue global de la base de données fédérée. L'instruction `COMMENT ON` est valide avec un pseudonyme et des colonnes définies par un pseudonyme. Cette instruction ne met pas à jour les catalogues de source de données.

Utilisation de pseudonymes dans les instructions GRANT et REVOKE

Les instructions `GRANT` et `REVOKE` sont valides avec un pseudonyme pour certains privilèges et pour tous les utilisateurs et groupes. Cependant, le système fédéré n'émet pas d'instruction `GRANT` ou `REVOKE` correspondante sur la source de données référencée par le pseudonyme.

Par exemple, supposons que l'utilisateur JEAN crée un pseudonyme pour une table Oracle n'ayant pas d'index. Le pseudonyme est ORAREM1. L'administrateur de bases de données Oracle définit par la suite un index pour cette table. L'utilisateur HELENE veut maintenant que la base de données fédérée sache que cet index existe afin que l'optimiseur de requêtes puisse définir des stratégies pour accéder à la table de façon plus efficace. HELENE peut informer la base de données fédérée qu'un nouvel index existe, en créant une spécification d'index pour ORAREM1.

Les informations relatives à l'index sont stockées dans la vue de catalogue SYSSTAT.INDEXES. Utilisez l'instruction GRANT pour accorder à HELENE les privilèges d'index sur ce pseudonyme pour qu'elle puisse créer la spécification d'index.

```
GRANT INDEX ON NICKNAME ORAREM1 TO USER HELENE
```

Pour révoquer les privilèges de création de spécification d'HELENE sur le pseudonyme ORAREM1, utilisez l'instruction REVOKE :

```
REVOKE INDEX ON ORAREM1 FROM USER HELENE
```

Impact des statistiques de sources de données sur les applications

Lorsqu'un pseudonyme est créé pour un objet de source de données, le catalogue global de la base de données fédérée est mis à jour avec les informations relatives à cet objet. L'optimiseur de requêtes utilise ces informations pour planifier comment extraire les données de l'objet.

Il est important de s'assurer que les informations de source de données sont à jour. La base de données fédérée ne détecte pas automatiquement les modifications apportées aux objets de source de données.

Statistiques d'objets de base de données stockées dans le catalogue global

Les informations relatives à un objet de source de données stockées dans le catalogue global dépendent du type d'objet. Pour les vues et tables de bases de données, le nom de l'objet, les attributs et les noms de colonnes sont stockés dans le catalogue global.

Dans le cas d'une table ou d'un pseudonyme, les informations stockées incluent également :

- Statistiques. Par exemple, le nombre de lignes et le nombre de pages sur lesquelles les lignes existent. Pour vous assurer que la base de données fédérée obtient les statistiques les plus à jour, exécutez l'équivalent de la commande RUNSTATS de la source de données sur la table avant de créer le pseudonyme.
- Descriptions d'index. Si la table ne comporte pas d'index, vous pouvez fournir au catalogue les métadonnées qu'une définition d'index contient généralement. Par exemple, admettons qu'un pseudonyme est créé pour une table distante et qu'un index est ensuite créé sur la table dans la source de données. Vous pouvez créer une spécification d'index sur le serveur fédéré représentant cet index distant. Créez une spécification d'index en émettant l'instruction CREATE INDEX et en référençant le pseudonyme de la table. Utilisez la clause SPECIFICATION ONLY avec l'instruction CREATE INDEX pour ne générer qu'une spécification d'index. La spécification d'index informe l'optimiseur fédéré qu'un index distant existe. Cependant, seules des métadonnées sont générées. Aucun index n'est réellement créé sur le serveur fédéré. De plus, aucune information statistique n'est fournie au catalogue global. Si vous donnez à la

spécification d'index exactement la même signature que l'index distant (c'est-à-dire le même nom et les mêmes colonnes dans le même ordre), vous pouvez utiliser SYSPROC.NNSTAT pour mettre à jour les statistiques de pseudonymes et de la spécification d'index.

Pour déterminer quelles sont les informations de source de données stockées dans le catalogue global, effectuez une requête sur les vues de catalogue SYSCAT.TABLES et SYSCAT.COLUMNS. Pour déterminer quelles sont les informations de source de données stockées dans le catalogue ou ce que contient une spécification d'index donnée, effectuez une requête sur la vue de catalogue SYSCAT.INDEXES.

Mise à jour des statistiques en utilisant la vue SYSSTAT au lieu de la vue SYSCAT

Les vues SYSCAT sont des vues de catalogue en lecture seule dans le schéma SYSCAT. Les vues SYSSTAT sont des vues de catalogue qui peuvent être mises à jour et qui contiennent des informations statistiques utilisées par l'optimiseur. Les vues SYSSTAT se trouvent dans le schéma SYSSTAT.

Si vous effectuez une opération UPDATE ou INSERT sur une vue du schéma SYSCAT, celle-ci échouera. Utilisez les vues de catalogues pouvant être mises à jour du schéma SYSSTAT pour modifier manuellement des statistiques relatives aux pseudonymes.

Définition des options de colonne sur les pseudonymes

Les options de colonne sont des paramètres des instructions CREATE NICKNAME et ALTER NICKNAME. Vous pouvez spécifier des options de colonne lorsque vous créez un pseudonyme ou en modifiant un pseudonyme existant.

Les informations que vous fournissez dans les options de colonne sont stockées dans le catalogue global.

Sources de données non relationnelles

Les options de colonne sont uniques pour chaque encapsuleur non relationnel. Ces options sont généralement définies lorsque vous émettez l'instruction CREATE NICKNAME.

Sources de données relationnelles

Il existe deux options de colonne que vous pouvez utiliser pour les sources de données relationnelles : NUMERIC_STRING et VARCHAR_NO_TRAILING_BLANKS.

Définition de l'option de colonne NUMERIC_STRING

Si une colonne de chaîne de source de données ne contient que des caractères numériques et aucun autre caractère (espace vide compris), définissez l'option de colonne NUMERIC_STRING sur Y.

Définir l'option de colonne NUMERIC_STRING sur Y permet d'optimiser les requêtes utilisant cette colonne pour les opérations de tri et de comparaison. Par exemple :

```
ALTER NICKNAME pseudonyme
  ALTER COLUMN nom_colonne_locale
  OPTIONS (SET NUMERIC_STRING 'Y')
```

Définition de l'option de colonne VARCHAR_NO_TRAILING_BLANKS

Si la colonne de chaîne de source de données ne contient pas de blanc de fin, définissez l'option de colonne VARCHAR_NO_TRAILING_BLANKS sur Y.

Certaines sources de données, Oracle par exemple, n'utilisent pas la logique de comparaison de chaîne espacée par un élément de remplissage qu'utilise la base de données fédérée. Cette remarque concerne les types de données tels que VARCHAR et VARCHAR2. De ce fait, les prédicats impliquant ces types de données doivent être réécrits par l'optimiseur de requêtes pour que les résultats de la requête soient cohérents.

La réécriture de ces instructions peut avoir un impact sur les performances. La définition de cette option pour une colonne spécifique permet de fournir à l'optimiseur de requêtes des informations relatives à ces colonnes pour qu'il puisse générer des instructions SQL plus efficaces.

Par exemple :

```
ALTER NICKNAME pseudonyme
  ALTER COLUMN nom_colonne_locale
  OPTIONS (SET VARCHAR_NO_TRAILING_BLANKS 'Y')
```

Chapitre 18. Création et utilisation des vues fédérées

Une vue qui inclut une référence à un pseudonyme dans l'instruction FULLSELECT est une *vue fédérée*. Les tables de base sont référencées dans la vue fédérée par des pseudonymes, au lieu du nom de la table de la source de données.

Restrictions

Les vues fédérées créées à partir de plusieurs objets de sources de données sont des vues en lecture seulement et ne peuvent pas être mises à jour.

Les vues fédérées créées à partir d'un seul objet de source de données peuvent être en lecture seulement ou pas.

- Une vue fédérée créée à partir d'une seule source de données non relationnelles est en lecture seulement.
- Une vue fédérée créée à partir d'une seule source de données relationnelles peut autoriser les mises à jours, selon le contenu de l'instruction CREATE VIEW.

A propos de cette tâche

Les avantages offerts par l'utilisation de vues fédérées sont identiques à ceux de l'utilisation de vues définies sur les tables locales d'un gestionnaire de bases de données relationnelles centralisé :

- Les vues fournissent une représentation intégrée des données
- Vous pouvez exclure d'une vue des colonnes de tables contenant des données sensibles ou confidentielles

Procédure

Vous créez une vue fédérée à partir d'objets de source de données ayant des pseudonymes. L'action de création d'une vue de base de données fédérée d'une source de données est parfois appelée «création d'une vue sur un pseudonyme». Cette expression illustre le fait que pour que la vue fédérée soit créée, l'instruction FULLSELECT CREATE VIEW doit référencer le pseudonyme de toutes les tables et vues de source de données que la vue fédérée doit contenir.

Création de vues fédérées - Exemples

Cette rubrique fournit des exemples de création de vues fédérées.

Exemple : Création d'une vue fédérée qui fusionne des données similaires provenant de plusieurs objets de source de données

Vous utilisez des données client sur trois serveurs distincts : un en Europe, un en Asie et un autre en Amérique du Sud. Les données client Europe se trouvent dans une table Oracle. Le pseudonyme de cette table est `ORA_EU_CUST`. Les données client Asie se trouvent dans une table Sybase. Le pseudonyme de cette table est `SYB_AS_CUST`. Les données client Amérique du Sud se trouvent dans une table Informix. Le pseudonyme de cette table est `INFMX_SA_CUST`. Chaque table a des colonnes contenant le numéro client (`CUST_NO`), le nom du client (`CUST_NAME`), le numéro du produit (`PROD_NO`) et la quantité commandée (`QUANTITY`). La syntaxe pour créer une vue à partir de ces trois pseudonymes fusionnant ces données client est :

```
CREATE VIEW FV1
  AS SELECT * FROM ORA_EU_CUST
  UNION
  SELECT * FROM SYB_AS_CUST
  UNION
  SELECT * FROM INFMX_SA_CUST
```

Exemple : Jointure de données pour la création d'une vue fédérée

Vous utilisez les données client sur un serveur et les données de ventes sur un autre serveur. Les données client se trouvent dans une table Oracle. Le pseudonyme de cette table est `ORA_EU_CUST`. Les données de ventes se trouvent dans une table Sybase. Le pseudonyme de cette table est `SYB_SALES`. Vous souhaitez faire correspondre les informations client avec les achats effectués par ces clients. Chaque table a une colonne contenant le numéro de client (`CUST_NO`). La syntaxe pour créer une vue fédérée à partir de ces deux pseudonymes fusionnant ces données est :

```
CREATE VIEW FV4
  AS SELECT A.CUST_NO, A.CUST_NAME, B.PROD_NO, B.QUANTITY
  FROM ORA_EU_CUST A, SYB_SALES B
  WHERE A.CUST_NO=B.CUST_NO
```

Chapitre 19. Préserver l'intégrité des données avec les niveaux d'isolement

Le niveau d'isolement définit le degré d'isolement d'un processus d'application des autres processus d'application exécutés simultanément.

Vous pouvez préserver l'intégrité des données d'une table de source de données en demandant le verrouillage des lignes de la table à un niveau d'isolement spécifique.

Le verrouillage s'effectue sur la ligne de la table de base de la source de données. Le gestionnaire de la base de données peut cependant remplacer plusieurs verrouillages de lignes par un verrouillage de table unique. Cette action s'appelle *l'escalade de verrous*. Un processus d'application obtient au minimum le niveau de verrouillage le plus bas demandé.

Les niveaux d'isolement pour la base de données fédérée sont les suivants :

- RR** Lecture reproductible
- RS** Lecture stable
- CS** Lecture non reproductible (niveau par défaut)
- UR** Lecture non validée

Les types d'isolement sont l'isolement au niveau de l'instruction et l'isolement au niveau de la connexion.

Vous pouvez définir l'isolement lorsque vous exécutez les actions suivantes :

- Précompilation ou liaison d'une application. Vous pouvez spécifier les niveaux d'isolement lorsque vous préparez ou liez une application. Le niveau d'isolement spécifié dans la commande BIND ou PREP est le niveau d'isolement par défaut lorsque le serveur fédéré se connecte à la source de données distante.
- Utilisation de la clause WITH dans une instruction SQL. Cette action s'appelle l'isolement au niveau de l'instruction. Vous pouvez utiliser la clause WITH dans les instructions SELECT, UPDATE, INSERT et DELETE.

Si le serveur fédéré ne trouve pas de niveau d'isolement pour une instruction, il utilise le niveau d'isolement défini lors de sa connexion à la source de données.

Le tableau suivant répertorie les sources de données utilisant l'isolement au niveau de la connexion, les niveaux d'isolement qu'elles utilisent et les niveaux d'isolement équivalents sur le serveur fédéré.

Tableau 21. Sources de données et niveaux d'isolement

Sources de données	Niveau d'isolement le plus restrictif	Niveau d'isolement plus restrictif	Niveau d'isolement moins restrictif	Niveau d'isolement le moins restrictif
Base de données fédérée	Lecture reproductible	Lecture stable	Lecture non reproductible	Lecture non validée
Famille de produits DB2	Lecture reproductible	Lecture stable*	Lecture non reproductible	Lecture non validée

Tableau 21. Sources de données et niveaux d'isolement (suite)

Sources de données	Niveau d'isolement le plus restrictif	Niveau d'isolement plus restrictif	Niveau d'isolement moins restrictif	Niveau d'isolement le moins restrictif
Informix	Lecture reproductible	Lecture reproductible	Lecture non reproductible	Lecture de pages modifiées
JDBC	Sérialisable	Lecture reproductible	Lecture validée	Lecture non validée
Microsoft SQL Server	Sérialisable	Lecture reproductible	Lecture validée	Lecture non validée
ODBC	Sérialisable	Lecture reproductible	Lecture validée	Lecture non validée
Oracle	Sérialisable	Sérialisable	Lecture validée	Lecture validée
Sybase	Niveau 3	Niveau 3	Niveau 1	Niveau 0

*Pour les sources de données DB2 UDB for VM and VSE Server, le niveau d'isolement est Lecture reproductible.

Le registre spécial CURRENT ISOLATION n'est pas utilisé par le serveur fédéré lorsqu'il se connecte à une source de données.

Les sources de données non relationnelles n'ont pas de concept correspondant aux niveaux d'isolement. OLE DB et Teradata ont un concept de niveaux d'isolement mais ceux-ci ne sont pas pris en charge par le serveur fédéré. Il n'existe pas de mappage de niveau d'isolement entre les niveaux d'isolement du serveur fédéré et les niveaux d'isolement des sources de données OLE DB, Teradata et des sources de données non relationnelles.

Isolement au niveau de l'instruction dans un système fédéré

Pour les sources de données fédérées, vous devez utiliser la clause d'isolement WITH pour spécifier l'isolement d'une instruction.

Vous devez utiliser la clause d'isolement WITH dans votre instruction si vous voulez utiliser l'isolement au niveau de l'instruction. Si vous utilisez des attributs avec l'interface CLI (Call Level Interface) ou d'autres API d'application pour l'isolement au niveau de l'instruction, ce dernier n'est pas affecté.

Les sources de données prenant en charge l'isolement au niveau de l'instruction dans un système fédéré sont les produits de la famille DB2 et Microsoft SQL Server. L'isolement d'instruction est envoyé aux sources de données éloignées pour la famille de produits DB2 et SQL Server.

Utilisez l'option de serveur DB2_STATEMENT_ISOLATION pour activer ou désactiver l'isolement au niveau de l'instruction. Vous pouvez spécifier cette option dans les instructions CREATE SERVER et ALTER SERVER. L'option de serveur est automatiquement définie sur Y.

Vous pouvez utiliser la clause d'isolement dans les instructions suivantes :

```
SELECT
SELECT INTO
Searched DELETE
INSERT
```

```
Searched UPDATE
DECLARE CURSOR
```

Clause de demande de verrouillage

Vous pouvez utiliser la clause de demande de verrouillage dans une instruction `SELECT` ou `SELECT INTO`. Les sources de données fédérées prenant en charge la clause de demande de verrouillage sont les sources de données DB2 for Linux, UNIX and Windows, ainsi que DB2 for z/OS.

Limitations de l'utilisation de la clause `WITH` pour définir le niveau d'isolement

Les conditions suivantes s'appliquent aux niveaux d'isolement spécifiés pour les instructions :

- La clause `WITH` ne peut pas être utilisée dans des sous-requêtes.
- Le niveau d'isolement `UR` ne s'applique que si la table de résultats de l'instruction `FULLSELECT` ou `SELECT INTO` est en lecture seule. Dans d'autres cas, le niveau d'isolement `UR` de l'instruction passe de `UR` à `CS` pour la famille de sources de données DB2. Pour la source de données SQL Server, le niveau d'isolement `UR` est promu au niveau lecture validée.
- Si vous spécifiez la clause de demande de verrouillage pour les sources de données suivantes, la clause est ignorée par le serveur fédéré.
 - DB2 for System i
 - DB2 for VM
 - Microsoft SQL Server

Isolement au niveau de la connexion dans un système fédéré

Le serveur fédéré mappe votre niveau d'isolement à un niveau correspondant sur la source de données.

Pour chaque connexion à la source de données, l'encapsuleur détermine le niveau d'isolement.

Lorsque le serveur fédéré se connecte à la source de données, le niveau d'isolement de la source de données distante est défini sur un niveau équivalent à celui du serveur fédéré. S'il n'existe pas d'équivalent exact, le serveur fédéré définit le niveau d'isolement sur le niveau de restriction suivant. Lorsqu'une connexion à la source de données est effectuée, le niveau d'isolement ne peut pas être modifié pendant toute la durée de la connexion.

Tous les encapsuleurs, à l'exception de Teradata, effectuent un suivi du niveau d'isolement. Lors de la définition d'une connexion, les encapsuleurs définissent son niveau d'isolement sur l'équivalent du niveau d'isolement actuel. Le niveau d'isolement actuel correspond à celui de la section en cours (première instruction fédérée d'une source de données). L'encapsuleur Teradata a toujours le niveau d'isolement lecture (niveau d'isolement par défaut) car il ne peut pas modifier le niveau d'isolement de la connexion.

Chapitre 20. Prise en charge des objets LOB fédérés

Dans un système de bases de données fédérées, vous pouvez accéder à des objets LOB sur des sources de données distantes et les manipuler.

Un système fédéré prend en charge les opérations SELECT sur les objets LOB dans des sources de données DRDA, Informix, Microsoft SQL Server, Oracle et Sybase. Par exemple :

```
SELECT nomemp, image FROM table_emp_infmx  
WHERE numemp = '01192345'
```

où *image* représente une colonne LOB et *table_emp_infmx* un pseudonyme référant une table Informix contenant des données relatives à l'employé.

Un système fédéré prend en charge les opérations SELECT, INSERT, UPDATE et DELETE sur les objets LOB dans les sources de données suivantes, en utilisant l'encapsuleur DRDA :

- DB2 for z/OS (version 7 ou supérieure)
- DB2 for System i (version 5)
- DB2 Database pour Linux, UNIX et Windows (Version 7 ou supérieure)

Les opérations de lecture et d'écriture prises en charge par DB2 Database pour Linux, UNIX et Windows sont répertoriées dans le tableau suivant :

Tableau 22. Prise en charge de la lecture et de l'écriture pour les objets LOB

Source de données	Type d'opération
DB2 for z/OS, DB2 for System i, DB2 Database pour Linux, UNIX et Windows ¹	lecture et écriture
BioRS	lecture seule
Informix	lecture seule
JDBC	lecture seule
Microsoft SQL Server	lecture seule
Oracle (encapsuleur NET8) ²	lecture et écriture
ODBC	lecture seule
Sybase	lecture seule
Teradata	lecture seule
services Web	lecture seule et liaison en sortie pour les objets CLOB uniquement
XML	lecture seule

Remarque :

1. DB2 for System i (version 5 ou ultérieure) est requis pour la prise en charge des objets LOB. DB2 Information Integrator Version 8 ne peut pas accéder aux données LOB DB2 Database pour Linux, UNIX et Windows Version 7.
 2. Pour exécuter des opérations d'insertion, de mise à jour et de suppression sur des colonnes Oracle LONG, vous devez migrer les colonnes LONG en objets LOB et recréer les pseudonymes.
-

Objets LOB Teradata

Les objets LOB Teradata sont légèrement différents des objets LOB DB2. Teradata n'a pas de types de données aussi grands que les objets LOB pris en charge dans DB2 pour Linux, UNIX et Windows. Cependant, il existe des types de données Teradata pouvant comporter jusqu'à 64 000 octets. Ces types de données sont : CHAR, VARCHAR, BYTE, VARBYTE, GRAPHIC et VARGRAPHIC. Ces types de données Teradata sont mappés à des types de données d'objets LOB DB2 lorsque la longueur du type de données Teradata dépasse les limites du type de données DB2 correspondant.

Longueurs des objets LOB

Certaines sources de données, comme Oracle et Informix, ne stockent pas les longueurs des colonnes LOB dans leurs catalogues système. Lorsque vous créez un pseudonyme sur une table, des informations provenant du catalogue système de la source de données sont extraites, dont la longueur de colonne. Comme aucune longueur n'existe pour les colonnes LOB, la base de données fédérée suppose que cette longueur est la longueur maximale d'une colonne LOB dans DB2 Database pour Linux, UNIX et Windows. La base de données fédérée stocke la longueur maximale dans le catalogue de base de données fédérée comme longueur de la colonne de pseudonyme.

Releveurs de coordonnées LOB

Les applications peuvent faire appel à des releveurs de coordonnées LOB pour obtenir des objets LOB stockés dans des sources de données distantes. Un releveur de coordonnées LOB est une valeur à 4 octets stockée dans une variable hôte. Une application peut utiliser le releveur de coordonnées LOB pour faire référence à une valeur LOB (ou expression LOB) contenue dans le système de base de données.

En utilisant un releveur de coordonnées LOB, une application peut manipuler la valeur LOB comme si elle était stockée dans une variable hôte classique. Lorsque vous utilisez des releveurs de coordonnées LOB, il n'y a pas besoin de transporter la valeur LOB du serveur source de données à l'application (et éventuellement ensuite de l'application au serveur).

La base de données fédérée peut extraire des objets LOB de sources de données distantes, les stocker sur le serveur fédéré puis émettre un releveur de coordonnées LOB sur l'objet LOB stocké. Les releveurs de coordonnées LOB sont libérés lorsque :

- Les applications émettent des instructions SQL FREE LOCATOR
- Les applications émettent des instructions COMMIT
- L'instance fédérée est redémarrée

Restrictions sur les objets LOB

Les systèmes fédérés imposent certaines restrictions sur les objets LOB.

Les restrictions suivantes s'appliquent aux objets LOB :

- La base de données fédérée ne peut pas lier des objets LOB distants à une variable de référence de fichier
- Les objets LOB ne sont pas pris en charge dans les sessions passe-système
- Les objets LOB ne sont pas pris en charge en tant que paramètres de procédures mémorisées

Considérations de performances pour le traitement des objets LOB

Lors du développement d'applications fédérées qui extraient et traitent des données LOB, les concepteurs d'applications et les administrateurs de bases de données doivent comprendre comment le traitement des objets LOB affecte les performances.

Lorsqu'une application extrait des données d'une source de données fédérée, le serveur fédéré doit extraire les données dans sa propre mémoire tampon avant d'envoyer les données à l'application. Comme les objets LOB ne sont pas traités dans un pool de mémoire tampon, les données LOB doivent d'abord passer par un espace table temporaire défini pour le serveur fédéré. Pour améliorer les performances et réduire la quantité de ressources utilisées, les concepteurs d'applications ne doivent matérialiser les données LOB que lorsque cela est nécessaire.

De même, lorsque le serveur fédéré met à jour des données LOB distantes, les données doivent passer par un espace table temporaire affecté au serveur fédéré avant d'être transmises à la source de données.

Les objets LOB en transit utilisent l'espace table temporaire affecté au serveur fédéré. De ce fait, les administrateurs de bases de données doivent augmenter la taille de cet espace table temporaire pour s'assurer qu'il est suffisant pour le traitement des objets LOB.

Recommandation : pour optimiser les performances lors de l'utilisation d'objets LOB, définissez l'espace table temporaire en tant qu'espace table SMS et vérifiez que celui-ci se trouve sur des disques à large bande passante E/S.

Utilisation de l'interface CLI DB2 pour accéder aux objets LOB fédérés

Le serveur fédéré prend en charge deux API CLI DB2 pour la sélection de données LOB :

- L'API SQLFetch extrait les objets LOB du serveur fédéré ou de la source de données dans les mémoires tampon en une seule opération.
- L'API SQLGetData extrait l'objet LOB par morceaux et peut nécessiter plusieurs appels pour extraire l'intégralité de l'objet LOB dans la mémoire tampon de l'application.

Recommandation : pour optimiser les performances, utiliser l'API SQLGetData lorsque vous extrayez des objets LOB via un serveur fédéré.

Le serveur fédéré prend en charge les API SQLExecute et SQLPutData pour la mise à jour des données LOB. L'API SQLExecute met à jour les données LOB en une seule opération, alors que l'API SQLPutData peut nécessiter plusieurs appels pour envoyer l'intégralité des données LOB des mémoires tampon de l'application au serveur. Chaque API s'exécute au même niveau dans un environnement fédéré.

Encapsuleurs sécurisés et isolés

Les pseudonymes créés pour les encapsuleurs définis comme étant sécurisés ou isolés ont les mêmes performances pour l'extraction ou la mise à jour d'objets LOB.

Chapitre 21. Requêtes réparties pour effectuer des requêtes sur des sources de données

Les requêtes soumises à la base de données fédérée peuvent demander des résultats d'une seule source de données mais il s'agit généralement de requêtes intégrant plusieurs sources de données. Dans la mesure où une requête classique est répartie sur plusieurs sources de données, elle est appelée *requête répartie*.

Une requête répartie utilise généralement une ou plusieurs des trois conventions SQL pour spécifier où les données doivent être extraites à partir de sous-requêtes, pour définir des opérateurs et pour joindre des sous-requêtes.

Requêtes réparties pour effectuer des requêtes sur des sources de données - exemples

Les exemples de cette rubrique illustrent plusieurs types de requêtes réparties : avec une sous-requête ; avec des opérateurs ensemblistes ; en vue d'une opération de jointure.

Dans cet exemple, le serveur fédéré est configuré pour accéder à une source de données DB2 for z/OS, à une source de données DB2 for System i et à une source de données Oracle. Une table contenant des informations relatives aux employés est stockée dans chaque source de données. Le serveur fédéré référence ces tables avec des pseudonymes pointant vers l'emplacement où se trouvent les tables.

zOS_EMPLOYEES

Pseudonyme d'une table se trouvant sur une source de données DB2 for z/OS et contenant des informations relatives aux employés.

SYSTEMi_EMPLOYEES

Pseudonyme d'une table se trouvant sur une source de données DB2 for System i et contenant des informations relatives aux employés.

ORA_EMPLOYEES

Pseudonyme d'une table se trouvant sur une source de données Oracle et contenant des informations relatives aux employés.

ORA_REGIONS

Pseudonyme d'une table se trouvant sur une source de données Oracle et contenant des informations relatives aux régions dans lesquelles vivent les employés.

Les exemples suivants illustrent les trois conventions SQL utilisées avec les requêtes réparties, en utilisant les pseudonymes définis pour chacune des tables.

Exemple : Requête répartie avec une sous-requête

La table SYSTEMi_EMPLOYEES contient les numéros de téléphones des employés vivant en Asie. Elle contient également les codes de région associés à ces numéros de téléphone mais ne répertorie pas les régions que ces codes représentent. ORA_REGIONS répertorie les codes et les régions. La requête suivante utilise une sous-requête pour rechercher le code de région pour la Chine. Elle utilise ensuite le code de région pour renvoyer la liste des employés de la table SYSTEMi_EMPLOYEES ayant un numéro de téléphone en Chine.

```
SELECT nom, téléphone FROM admindb2.employés_SYSTEMi
WHERE code_région IN
(SELECT code_région FROM adminbd.régions_ora
WHERE nom_région = 'CHINE')
```

Exemple : Requête répartie avec des opérateurs ensemblistes

Le serveur fédéré prend en charge trois opérateurs ensemblistes : UNION, EXCEPT et INTERSECT.

- Utilisez l'opérateur ensembliste UNION pour combiner des lignes correspondant à l'une de deux (ou plus) instructions SELECT.
- Utilisez l'opérateur ensembliste EXCEPT pour extraire les lignes correspondant à la première instruction SELECT mais pas à la seconde.
- Utilisez l'opérateur ensembliste INTERSECT pour extraire les lignes correspondant aux deux instructions SELECT.

Ces trois opérateurs ensemblistes peuvent utiliser l'opérande ALL pour indiquer que les lignes en double ne doivent pas être supprimées des résultats. Cela supprime le besoin d'un tri supplémentaire.

La requête suivante extrait tous les noms d'employés et tous les codes de régions présents dans les deux tables, SYSTEMi_EMPLOYEES et zOS_EMPLOYEES, même si ces tables se trouvent sur des sources de données différentes.

```
SELECT nom, code_région
FROM employés_as400
INTERSECT
SELECT nom, code_région
FROM employés_zOS
```

Exemple : Requête répartie pour une jointure

Une jointure relationnelle produit un ensemble de résultats contenant une combinaison de colonnes extraites de deux tables. Vous devez spécifier des conditions pour limiter la taille des lignes dans l'ensemble de résultats.

La requête ci-dessous combine les noms d'employés et les noms des régions correspondantes en comparant les codes de région répertoriés dans les deux tables. Chaque table se trouve sur une source de données différente.

```
SELECT nom.t1, nom_région.t2
FROM adminbd.employés_SYSTEMi t1, adminbd.régions_ora t2
WHERE code_région.t1 = code_région.t2
```

Optimisation des requêtes réparties avec des options de serveur

Dans un système fédéré, utilisez des paramètres appelés *options de serveur* pour communiquer au catalogue global les informations s'appliquant à la source de données dans son intégralité ou pour contrôler l'interaction entre la base de données fédérée et une source de données.

A propos de cette tâche

Les options de serveur décrivent les fonctions d'une source de données spécifique et développent les connaissances dont dispose le serveur fédéré à propos de cette source de données. Par exemple, vous pouvez utiliser les options de serveur suivantes :

- L'option de serveur `VARCHAR_NO_TRAILING_BLANKS` informe l'optimiseur que chaque colonne `VARCHAR` se trouvant sur ce serveur de source de données n'a pas de blanc de fin. N'utilisez cette option que lorsque vous êtes certain qu'aucune des colonnes `VARCHAR2` de tous les objets référencés par un pseudonyme sur ce serveur n'a un blanc de fin. Sinon, utilisez une option de colonne pour spécifier les colonnes d'objets spécifiques sur ce serveur qui n'ont pas de blanc de fin. L'option de colonne est également appelée `VARCHAR_NO_TRAILING_BLANKS`.
- L'option de serveur `PLAN_HINTS` communique aux sources de données Sybase des fragments d'instruction, appelés *indications de plan*. Les indications de plan peuvent aider l'optimiseur de source de données à décider quel index utiliser pour accéder à une table et quelle séquence de jointure de table utiliser pour extraire les données d'un ensemble de résultats.

Généralement, les administrateurs de bases de données définissent des options de serveur pour un système fédéré. Un programmeur peut cependant tirer parti des options de serveur pour optimiser les requêtes. Par exemple, pour les sources de données `SYB1` et `SYB2`, l'option de serveur `PLAN_HINTS` est définie sur la valeur par défaut, `N` (non, ne pas fournir d'indication de plan à cette source de données). Vous écrivez une requête répartie qui sélectionne des données dans `SYB1` et `SYB2`. Vous vous attendez à ce que les optimiseurs de ces sources de données puissent utiliser les indications de plan pour améliorer leurs stratégies d'accès à ces données. Vous pouvez remplacer la valeur par défaut par la valeur `Y` (oui, fournir des indications de plan) pendant que votre application est connectée à la base de données fédérée. Lorsque la connexion à la source de données prend fin, le paramètre est automatiquement réinitialisé sur la valeur par défaut, `N`.

Procédure

Pour définir des options de serveur :

Utilisez l'instruction `SET SERVER OPTION` pour définir ou modifier des options de serveur. Pour vous assurer que le paramètre prend effet, spécifiez l'instruction `SET SERVER OPTION` immédiatement après l'instruction `CONNECT`. L'option de serveur est effective pendant la durée de la connexion à la base de données fédérée.

Envisagez de préparer l'instruction de façon dynamique. L'instruction `SET SERVER OPTION` n'affecte que les instructions SQL dynamiques.

En SQL statique, l'instruction SET SERVER OPTION affecte uniquement l'exécution de l'instruction SQL statique. Elle n'a aucun impact sur les plans générés par l'optimiseur.

Annulation d'une requête fédérée

Vous pouvez interrompre une application et annuler une requête. Avec ce support, vous pouvez annuler une requête sur l'application de source de données distante avant qu'elle ne se termine et propager l'interruption et l'annulation de la requête à la source de données distante.

Restrictions

Dans la version 9.7, vous pouvez interrompre une application et annuler une requête sur la source de données DB2 Database for Linux, UNIX ou Windows uniquement.

Vous pouvez annuler les instructions fédérées SELECT, les instructions INSERT, UPDATE et DELETE, les procédures stockées fédérées et les instructions dans les sessions passe-système.

Une instruction fédérée peut contenir plusieurs segments qui accèdent à plusieurs sources de données. Si un segment accède à une source de données qui n'est pas prise en charge (plusieurs segments de requête sur la source de données Sybase, par exemple) vous ne pouvez pas annuler l'instruction.

Vous ne pouvez pas annuler une requête fédérée lorsque ATQ (Asynchronous Table Queue) est activé.

A propos de cette tâche

Lorsque le serveur fédéré exécute une instruction SQL fédérée et qu'il est bloqué en attendant le résultat ou une réponse de la source de données distante, l'application sur le serveur fédéré a l'état 'En attente d'une réponse fédérée'. Vous pouvez exécuter la commande LIST APPLICATIONS pour identifier les applications ayant l'état 'En attente d'une réponse fédérée'.

Lorsqu'une application a cet état, vous pouvez utiliser la séquence de touches Ctrl-C pour interrompre et annuler l'application ou utiliser la commande FORCE APPLICATION pour arrêter une application sur le serveur distant.

Procédure

Pour annuler une requête fédérée, procédez de l'une des manières suivantes :

- Appuyez sur Ctrl-C pour interrompre l'application.

Vous pouvez appuyer sur Ctrl-C pour interrompre l'application et annuler la requête sur le serveur distant. L'instruction SQL en cours sur le serveur fédéré est interrompue et annulée et la cohérence de la base de données locale est maintenue. L'instruction distante est également annulée. Les connexions entrantes et sortantes restent actives.

- Exécutez la commande FORCE APPLICATION

Vous pouvez exécuter la commande FORCE APPLICATION pour arrêter une application. L'application est arrêtée sur le serveur fédéré et la cohérence de la transaction est maintenue localement et sur le serveur distant. Les connexions entrantes et sortantes sont fermées et la partie distante de l'exécution de l'application sur la source de données distante est annulée.

Chapitre 22. Interrogation des sources de données avec des sessions passe-système

Utilisez des sessions passe-système pour exécuter des opérations qui ne sont pas réalisables avec SQL/API DB2.

A propos de cette tâche

Les session passe-système sont utiles lorsque :

- Les applications doivent créer des objets sur la source de données ou exécuter des opérations INSERT, UPDATE ou DELETE.
- La base de données fédérée ne prend pas en charge une opération de source de données unique.

Procédure

Pour effectuer directement une requête de sources de données avec la fonction passe-système :

- Utilisez l'instruction SET PASSTHRU pour démarrer une session passe-système et accéder directement à un serveur. Vous pouvez émettre cette instruction dynamiquement. Voici un exemple de cette instruction : SET PASSTHRU ORACLE1 Cette instruction SET PASSTHRU ouvre une session passe-système vers la source de données utilisant le nom de serveur ORACLE1. ORACLE1 est le nom que vous avez enregistré pour le serveur de source de données lorsque vous avez créé la définition de serveur.
- Lorsque la session passe-système est ouverte, vérifiez que vous utilisez le nom réel de l'objet et non pas le pseudonyme lorsque vous référencez des objets dans une session passe-système. Vous devez utiliser le dialecte SQL de la source de données, à moins que la base de données fédérée soit la source de données référencée.
- Si une instruction statique est émise au cours d'une session passe-système, elle est envoyée au serveur fédéré pour traitement. Si vous voulez soumettre une instruction SQL pour traitement à une source de données, vous devez la préparer dynamiquement dans la session passe-système et elle doit être exécutée pendant que la session est encore ouverte. Pour préparer des instructions dynamiquement dans une session passe-système :
 - Pour soumettre une instruction SELECT, utilisez l'instruction PREPARE avec celle-ci, puis utilisez les instructions OPEN, FETCH et CLOSE pour accéder aux résultats de votre requête.
 - Pour une instruction prise en charge autre que SELECT, vous avez deux possibilités. Vous pouvez utiliser l'instruction PREPARE pour préparer l'instruction prise en charge et l'instruction EXECUTE pour l'exécuter. Vous pouvez également utiliser l'instruction EXECUTE IMMEDIATE pour préparer et exécuter l'instruction.

Si vous émettez la commande COMMIT ou ROLLBACK au cours d'une session passe-système, cette commande achèvera l'unité de travail en cours mais ne mettra pas fin à la session passe-système.

Considérations et restrictions relatives au passe-système fédéré

Cette rubrique décrit les considérations et les restrictions dont vous devez avoir connaissance lorsque vous utilisez une session passe-système.

Les considérations et restrictions suivantes s'appliquent à toutes les sources de données :

- Les instructions préparées au cours d'une session passe-système doivent être exécutées au cours de cette même session. Les instructions préparées au cours d'une session passe-système mais exécutées en dehors de cette session échouent et génèrent une erreur SQLSTATE 56098. Les instructions préparées en dehors d'une session passe-système mais exécutée dans une session passe-système sont gérées comme une instruction SET PASSTHRU.
- Une application peut émettre plusieurs instructions SET PASSTHRU mais seule la dernière session est active. Lorsqu'une nouvelle instruction SET PASSTHRU est appelée, elle met fin à l'instruction SET PASSTHRU précédente. Vous ne pouvez pas passer par plusieurs sources de données au cours d'une même session passe-système.
- Si plusieurs sessions passe-système sont utilisées dans une application, assurez-vous d'avoir émis une instruction COMMIT avant d'ouvrir une autre session passe-système. Cela permet de conclure l'unité de travail de la session en cours.
- Les marqueurs de paramètre ne sont pas pris en charge dans les sessions passe-système. Utilisez des variables hôtes à la place des marqueurs de paramètre.
- Vous pouvez utiliser la sémantique WITH HOLD sur un curseur défini dans une session passe-système. Vous risquez cependant de recevoir une erreur si vous tentez d'utiliser la sémantique WITH HOLD à la suite d'une instruction COMMIT et que la source de données ne prend pas en charge cette sémantique.
- Les variables hôtes définies dans les instructions SQL au cours d'une session passe-système doivent être au format :H*n* où H est en majuscule et où *n* est un entier unique. Les valeurs de *n* doivent être numérotées de façon consécutive en commençant par zéro.
- La fonction passe-système ne prend pas en charge les objets LOB.
- La fonction passe-système ne prend pas en charge les appels de procédure mémorisée.
- La fonction passe-système ne prend pas en charge l'instruction SELECT INTO.
- La fonction passe-système ne prend pas en charge les fonctions définies par l'utilisateur externe ou SQL.
- Vous ne pouvez pas exécuter d'instructions ROLLBACK ou SQL COMMIT dynamiques lors d'une session passe-système.
- Lorsque vous effectuez une opération de mise à jour ou de suppression au cours d'une session passe-système, vous ne pouvez pas utiliser la condition WHERE CURRENT OF CURSOR.
- En mode passe-système, les instructions SQL dynamiques sont exécutées à distance alors que les instructions SQL statiques sont envoyées au serveur fédéré pour traitement. Si vous voulez soumettre une instruction SQL pour traitement à une source de données, vous devez la préparer dynamiquement dans la session passe-système et elle doit être exécutée pendant que la session est encore ouverte.

- Les instructions SET PASSTHRU statiques dans les procédures SQL sont bloquées lorsque des procédures mémorisées sont créées et une erreur SQL0104N est émise. Pour activer et désactiver le mode passe-système, utilisez l'instruction EXECUTE IMMEDIATE.

Exemple :

```

create procedure stp()
dynamic result sets 1
language sql
modifies sql data
begin
declare stmt varchar(100);

DECLARE cur1 CURSOR WITH RETURN TO CALLER
FOR SELECT * FROM t1 ;

set stmt = 'set passthru mvs7';
execute immediate stmt;

set stmt = 'insert into t1 values (20, 'passthru_insert')';
execute immediate stmt;
commit;

insert into t1 values (20, 'stp_insert');
commit;

OPEN cur1;

end
DB20000I The SQL command completed successfully.

call stp()

```

```

Result set 1
-----
10 local
20 stp_insert

```

2 record(s) selected.

Return Status = 0

```

select * from t1

C1 C2
-----
10 remote
20 passthru_insert

```

2 record(s) selected.

```

set passthru reset
DB20000I The SQL command completed successfully.

```

```

select * from t1

C1 C2
-----
10 local
20 stp_insert

```

2 record(s) selected.

- Quitter une procédure mémorisée ou une instruction SQL composée ne met pas automatiquement fin au mode passe-système. Pour quitter le mode passe-système, vous devez appeler explicitement l'instruction SET PASSTHRU RESET, comme indiqué dans l'exemple ci-dessus.

Sessions passe-système vers des sources de données Oracle

Cette rubrique identifie certaines considérations SQL dont vous devez avoir connaissance avant de soumettre des instructions SQL à des sources de données Oracle au cours d'une session passe-système.

- Toutes les instructions DDL émises sur un serveur Oracle sont exécutées lors de l'analyse et ne sont pas soumises à la sémantique de transaction. L'opération, une fois terminée, est automatiquement validée par Oracle. Si une annulation est effectuée, l'instruction DDL n'est pas annulée.
- Lorsque vous émettez une instruction SELECT à partir de types de données brutes, utilisez la fonction RAWTOHEX pour recevoir les valeurs hexadécimales. Lorsque vous exécutez une instruction INSERT sur des types de données brutes, fournissez la représentation hexadécimale.

Chapitre 23. Réglage fin du traitement des requêtes

Vous pouvez régler le système fédéré afin d'améliorer les performances du traitement de la requête.

A propos de cette tâche

Lorsque vous soumettez des requêtes SQL à la base de données fédérée, le compilateur SQL traite la requête et l'optimiseur de requêtes l'analyse puis crée un plan d'accès. L'optimiseur de requêtes stocke les informations sur le plan d'accès dans les tables Explain de la base de données fédérée. Vous pouvez utiliser les outils de format de table Explain, db2expln et dynexpln pour comprendre le plan d'accès pour une instruction SQL particulière.

Procédure

Pour régler les systèmes fédérés afin d'optimiser le traitement des requêtes, procédez comme suit :

1. Soumettez les requêtes SQL que vous souhaitez régler sur la base de données fédérée.
2. Analysez l'emplacement où la requête est évaluée.
3. Etudiez les raisons des décisions prises dans le plan d'accès et modifiez votre système pour augmenter les chances de transfert.

Publications sur les performances des données fédérées

Vous pouvez vous référer à de nombreux documents IBM qui comportent des informations détaillées sur les réglage des performances.

- Exécution asynchrone des requêtes fédérées dans WebSphere Federation Server V9.1, sur <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0611norwood/>
- Optimiser les performances de WebSphere Information Integrator avec les tables de requêtes fédérées, sur <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0605lin/index.html>
- Amélioration des performances dans IBM WebSphere Federation Server V9.1, Partie 1 : Améliorer les performances des requêtes fédérées avec les nouvelles fonctions de WebSphere Federation Server, sur <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert/>
- Amélioration des performances dans IBM WebSphere Federation Server V9.1, Partie 2 : Caractéristiques des performances des nouvelles fonctionnalités dans WebSphere Federation Server, sur <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert2/>
- Utilisation de la technologie de fédération de données dans IBM WebSphere Information Integrator : Exemples d'utilisation de la fédération de données et réglage des performances, sur <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0507lin/>
- Parallélisme dans WebSphere Information Integrator V8.2, sur <http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0502harris/>

- Fédération de données avec IBM DB2 Information Integrator V8.1, sur <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247052.html?Open>
- "Utilisation de la technologie de base de données fédérée de IBM DB2 Information Integrator", sur <ftp://ftp.software.ibm.com/software/data/pubs/papers/iifed.pdf>

Analyse de requête

L'analyse qui détermine le réglage à appliquer à la requête pour optimiser les performances est une partie importante du traitement de requête.

Pour obtenir des données des sources de données, les clients (utilisateurs et applications) soumettent des requêtes en langage SQL à la base de données fédérée. Le compilateur SQL consulte ensuite les informations dans le catalogue global et l'encapsuleur de source de données pour l'aider à traiter la requête. Il comprend les informations relatives à la connexion à la source de données, aux attributs de serveur, aux mappages, aux informations d'indexation et aux statistiques de pseudonymes.

Dans le cadre du processus du compilateur SQL, l'*optimiseur de requête* analyse une requête. Le compilateur développe des stratégies alternatives, appelées *plans d'accès*, pour traiter la requête. Les plans d'accès peuvent lancer la requête pour :

- Traitement par les sources de données
- Traitement par le serveur fédéré
- Traitement en partie par les sources de données et en partie par le serveur fédéré

La base de données fédérée évalue les plans d'accès essentiellement sur la base des informations relatives aux capacités de la source de données et aux attributs des données. Ces informations se trouvent dans l'encapsuleur et le catalogue global. La base de données fédérée décompose la requête en segments appelés *fragments de requête*. En règle générale, il est plus efficace de transférer l'exécution d'un fragment de requêtes sur une source de données, si la source de données peut traiter le fragment. Toutefois, l'optimiseur de requêtes prend en compte d'autres facteurs tels que :

- Le volume de données à traiter.
- La vitesse de traitement de la source de données.
- Le volume de données qui sera renvoyé par le fragment.
- La bande passante de la communication.

L'analyse du transfert de l'exécution sur la base de données s'effectue uniquement sur des sources de données relationnelles. Les sources de données non relationnelles utilisent le protocole RRC (Request-Reply-Compensate).

La figure suivante illustre les étapes effectuées par le compilateur SQL quand il traite une requête.

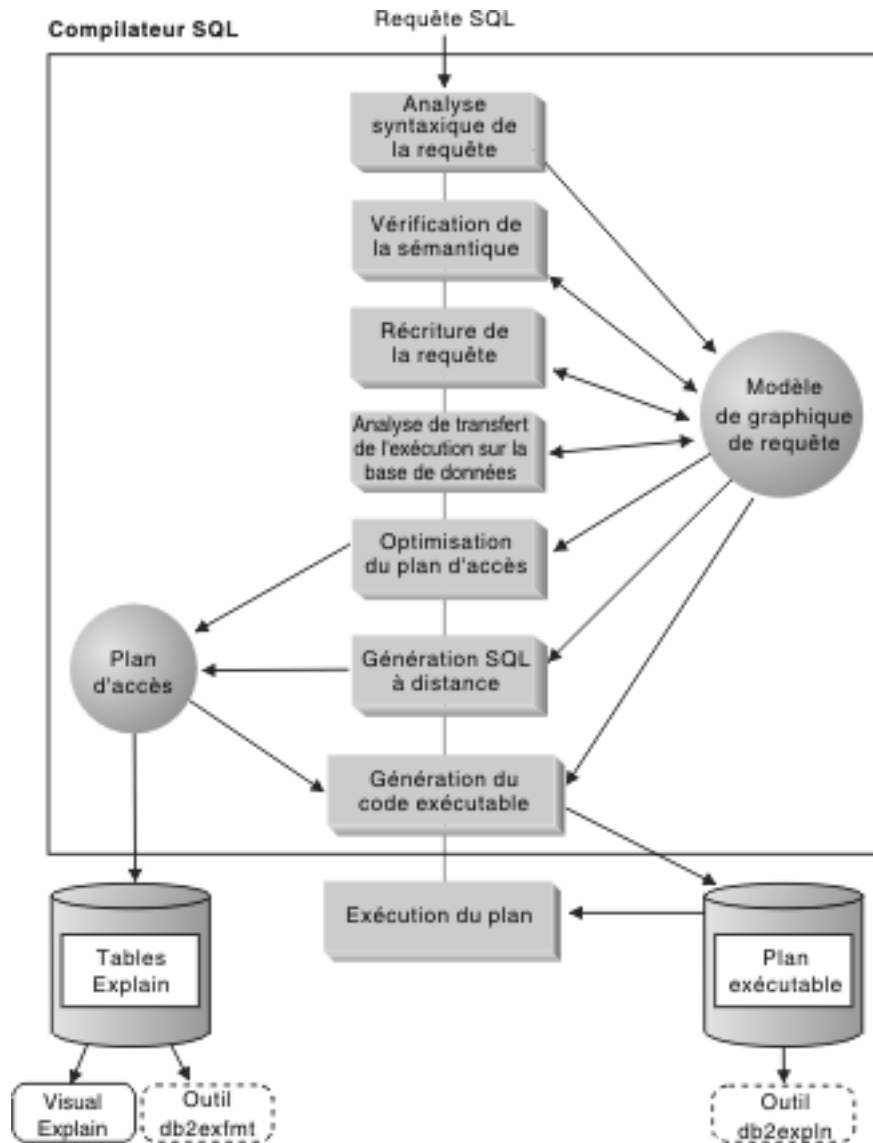


Figure 9. Graphique de flux de l'analyse de la requête du compilateur SQL

L'optimiseur de requête génère des plans d'accès locaux et éloignés pour le traitement d'un fragment de requête, en fonction du coût de la ressource. La base de données fédérée choisit ensuite le plan qu'elle considère comme le plus économique en termes de ressource dans le traitement d'une requête.

Si un des fragments doit être traité par des sources de données, le serveur fédéré soumet ces fragments aux sources de données. Après le traitement des fragments par les sources de données, les résultats sont extraits et renvoyés au serveur fédéré. Si la base de données fédérée effectue une partie du traitement, elle associe ses résultats à ceux extraits de la source de données. Le serveur fédéré renvoie ensuite les résultats au client.

La tâche principale de l'analyse de transfert de l'exécution sur la base de données est d'identifier les opérations pouvant être évaluées à distance. Pour ce faire, elle se base sur l'instruction SQL qu'elle reçoit et sur ses connaissances des fonctions et de la sémantique de la source de données éloigné. A partir de cette analyse, l'optimiseur de requête évalue les alternatives puis choisit le plan d'accès en

fonction du coût. L'optimiseur peut choisir de ne pas effectuer une opération directement sur une source de données éloignée si elle n'est pas rentable. La seconde tâche correspond à la tentative de récrire la requête pour compenser la différence de sémantique et d'opérations SQL entre le serveur fédéré et la source de données afin d'optimiser la requête.

Le plan d'accès final sélectionné par l'optimiseur peut inclure des opérations évaluées sur les sources de données éloignées. Pour ces opérations effectuées à distance, le compilateur SQL crée un langage SQL efficace dans le dialecte SQL de la source de données éloignée au cours de la phase de génération. Le processus de production d'un plan de requête optimal prenant en compte toutes les sources s'appelle *l'optimisation globale*.

Pour les sources non relationnelles, les encapsuleurs utilisent le protocole RRC (Request-Reply-Compensate).

Analyse du transfert de l'exécution sur la base de données

L'analyse de transfert de l'exécution sur la base de données informe l'optimiseur de requête si une source de données éloignée peut effectuer une opération. Une opération peut être une fonction, telle qu'un opérateur relationnel, des fonctions utilisateur ou système, ou un opérateur SQL (GROUP BY, ORDER BY, entre autres). L'optimiseur s'appuie ensuite sur le coût pour décider s'il transfère ou non l'exécution de l'opérateur sur la base de données. Même si l'analyse de transfert de l'exécution sur la base de données détermine qu'une opération particulière peut être effectuée sur la source éloignée, l'optimiseur peut décider de l'exécuter localement sur le serveur fédéré, s'il s'avère qu'elle sera plus économique en termes de ressources.

L'analyse du transfert de l'exécution sur la base de données s'effectue sur des sources de données relationnelles. Les sources non relationnelles utilisent le protocole RRC (Request-Reply-Compensate).

Les fonctions dont l'exécution ne peut être transférée, peuvent avoir un impact considérable sur les performances de la requête. Examinez l'effet du forçage de l'évaluation d'un prédicat sélectif au niveau local plutôt qu'au niveau de la source de données éloignée. Cette approche risque de nécessiter l'extraction par le serveur fédéré de la table entière de la source de données éloignée puis le filtrage de la table localement à l'aide du prédicat. Si votre réseau dispose de contraintes, et que la table est volumineuse, les performances de la requête risquent d'en souffrir.

Les opérateurs dont l'exécution n'est pas transférée peuvent également avoir une incidence considérable sur les performances de la requête. Par exemple, des données éloignées d'un agrégat d'opérateur GROUP BY en local peuvent, une fois de plus, nécessiter que le serveur fédéré puisse extraire la table entière de la source de données éloignée.

Par exemple, supposons que le pseudonyme EMP référence la table EMPLOYEE. Cette table contient 10000 lignes. Une colonne contient les codes postaux et une autre contient les salaires de chaque employé. La requête suivante est envoyée au serveur fédéré pour compter le nombre d'employés par ville qui gagnent plus de 50000 et qui vivent dans une catégorie de code postal particulière :

```
SELECT CITY, COUNT(*) FROM EMP
WHERE ZIP BETWEEN 'CA1' AND 'CA5' AND SALARY > 50000
GROUP BY CITY;
```


Quand le compilateur SQL reçoit cette instruction, il étudie plusieurs possibilités :

- Les séquences de classement de la source de données et du serveur fédéré sont identiques. Il est probable que l'exécution des deux prédicats sera transférée sur la base de données, car ils peuvent permettre de réduire la taille de l'ensemble de résultats intermédiaires de la source de données sur le serveur fédéré. En règle générale, il est plus rentable de filtrer et de grouper les résultats sur la source de données que de copier la totalité de la table sur le serveur fédéré et d'effectuer les opérations localement. L'analyse du transfert de l'exécution sur la base de données détermine si des opérations peuvent être effectuées sur la source de données. Les séquences de classement étant identiques, les prédicats et l'opération GROUP BY peuvent avoir lieu sur la source de données.
- Les séquences de classement sont identiques et l'optimiseur de requête sait que le serveur fédéré est très rapide. Il est possible que l'optimiseur de requête décide que la meilleure approche (la plus économique) consiste à exécuter l'opération GROUP BY localement. L'exécution des prédicats sera transférée sur la source de données pour évaluation. Voici un exemple d'analyse de transfert de l'exécution sur la base de données associée à l'optimisation globale.
- Les séquences de classement sont différentes. L'exécution du prédicat SALARY sera probablement transférée sur la source de données, car les colonnes numériques sont triées de la même manière, quelle que soit la séquence de classement. Toutefois, l'exécution du prédicat sur ZIP ne sera pas transférée car il dépend de l'ordre d'une colonne de caractère. L'exécution de l'opération GROUP BY ne sera pas transférée à moins que l'exécution des prédicats sur ZIP et SALARY le soit également.

Le compilateur SQL va étudier les plans d'accès disponibles puis choisir le plus rentable.

En règle général, l'objectif est de garantir que l'optimiseur de requête transfère l'exécution des fonctions et des opérateurs sur les sources de données pour évaluation. De nombreux facteurs peuvent avoir une incidence sur la possibilité ou non qu'une fonction ou un opérateur SQL soit évalué sur une source de données éloignée. Les principaux facteurs qui influencent l'optimiseur de requête sont les suivants : caractéristiques de serveur, caractéristiques de pseudonymes et caractéristiques de requête.

Caractéristiques de serveur ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données

Les caractéristiques de serveur qui ont une incidence sur le transfert de l'exécution sur la base de données incluent la prise en charge du langage SQL, la séquence de classement, les options de serveur fédéré et les mappages de type et de fonction.

Les facteurs qui influencent les possibilités de transfert de l'exécution des sources de données non relationnelles sont différents de ceux qui touchent les sources de données relationnelles. Le dialecte SQL n'est pas un facteur déterminant pour la plupart des sources de données non relationnelles, puisqu'elles n'utilisent pas le langage SQL.

Les rubriques suivantes décrivent les facteurs propres à la source de données qui peuvent être déterminants pour les possibilités de transfert de l'exécution sur la base de données.

Différences SQL

Les caractéristiques SQL qui ont une incidence sur le transfert de l'exécution sur la base de données incluent les fonctions SQL, les restrictions, les limitations et les éléments SQL propres au serveur.

- Fonctions SQL. Chaque source de données prend en charge une variante du dialecte SQL et différents niveaux de fonctionnalité. Par exemple, examinons la liste GROUP BY. La plupart des sources de données prennent en charge l'opérateur GROUP BY. Toutefois, certaines sources de données sont limitées dans le nombre d'éléments d'une liste GROUP BY ou dans l'acceptation ou non d'une expression dans la liste GROUP BY. Si des restrictions existent sur la source de données éloignée, le serveur fédéré risque d'effectuer l'opération GROUP BY localement.
- Restrictions SQL. Chaque source de données peut avoir différentes restrictions SQL. Par exemple, certaines sources de données nécessitent des marques de paramètres pour relier des valeurs aux instructions SQL éloignées. Par conséquent, vous devez vérifier les restrictions propres aux marques de paramètres pour garantir que chaque source de données prend en charge une telle méthode de liaison. Si le serveur fédéré ne peut pas déterminer une bonne méthode de liaison d'une valeur à une fonction, cette fonction doit être évaluée localement.
- Limitations SQL. Le serveur fédéré peut autoriser l'utilisation d'entiers plus grands par rapport aux sources de données. Toutefois, les valeurs qui dépassent les limites ne peuvent pas être imbriquées dans les instructions qui sont envoyées aux sources de données. Par conséquent, la fonction ou l'opérateur qui agit sur cette constante doit être évalué localement.
- Éléments spécifiques du serveur. Plusieurs facteurs entrent dans cette catégorie. Par exemple, le tri des valeurs NULL (de la plus grande ou la plus petite, en fonction de la commande). Par exemple, si la valeur NULL est triée sur une source de données d'une manière différente que sur le serveur fédéré, les opérations ORDER BY effectuées sur une expression à valeur NULL ne peuvent pas être évaluées à distance.

Type de données VARCHAR2 dans les systèmes fédérés

Pour régler le traitement de votre requête, votre plan d'accès doit tenir compte du remplissage de blancs avec des données VARCHAR2.

Vous utilisez l'option de serveur VARCHAR2_COMPAT pour activer la prise en charge des sources de données compatibles VARCHAR2.

La sémantique de compatibilité VARCHAR2 détermine la manière dont votre serveur fédéré et source de données traitent le type de données VARCHAR2 selon que le serveur fédéré, la source de données ou les deux sont compatibles avec VARCHAR2. Les systèmes incluent les configurations possibles suivantes :

- Votre serveur fédéré est compatible VARCHAR2, mais se connecte à une source de données qui n'est pas compatible VARCHAR2.
- Votre serveur fédéré n'est pas compatible VARCHAR2, mais se connecte à une source de données qui l'est.
- Votre serveur fédéré et source de données sont tous les deux compatibles VARCHAR2.

Lorsque votre serveur fédéré ou source de données DB2 est compatible VARCHAR2, le type de données VARCHAR2 est traité comme un synonyme pour le type de données VARCHAR.

Conseil : Pour garantir de meilleures performances, dans les systèmes qui se connectent uniquement à des sources de données compatibles VARCHAR2, votre serveur fédéré doit également être compatible VARCHAR2.

Dans les scénarios suivants, le serveur fédéré traite vos valeurs de chaîne vides et transmet les opérations vers la source de données en fonction de la sémantique de compatibilité VARCHAR2.

Scénario 1

Le serveur fédéré n'est pas compatible VARCHAR2, mais se connecte à une source de données qui l'est. Le serveur fédéré autorise les chaînes vides, mais pas la source de données. Ainsi, toutes les chaînes vides qui sont transmises à la source de données sont converties en valeurs NULL.

Par défaut, le serveur fédéré utilise une sémantique de remplissage de blancs sur le type de données CHAR avant de transmettre le résultat à la source de données. Cependant, le serveur fédéré n'utilise pas le remplissage de blancs dans les opérations qui spécifient uniquement des chaînes vides.

Exemples

- Les instructions INSERT et UPDATE suivantes n'incluent que des chaînes vides non remplies de blancs. Les instructions transmettent les valeurs de chaîne vides qui sont converties en valeurs NULL par la source de données.

Par exemple, les instructions suivantes ne sont pas remplies de blancs par le serveur fédéré :

```
INSERT INTO n1 (c1) VALUES ('')
```

```
UPDATE n1 SET c1=''
```

```
INSERT INTO n1 (c1) VALUES (''), ('')
```

- L'instruction INSERT suivant envoie 10 blancs à la source de données au lieu de la chaîne vide :

```
INSERT INTO n1 (col_char) VALUES (''), ('ibm')
```

Scénario 2

Le serveur fédéré est compatible VARCHAR2, mais se connecte à une source de données qui ne l'est pas. La sémantique de compatibilité VARCHAR2 du serveur fédéré n'autorise pas les chaînes vides et qui n'utilise pas la sémantique de remplissage de blancs. Toutefois, la sémantique de la source de données autorise les chaînes vides et utilise la sémantique de remplissage de blancs. Ainsi, le serveur fédéré empêche le transfert des opérations impliquant des chaînes vides, sauf si la sémantique et la cohérence des données du serveur sont préservées.

Exemples

- L'instruction INSERT suivante est transmise à la source de données, car la chaîne vide est convertie en valeur NULL et ne peut pas être traitée localement :

```
INSERT INTO n1 (c1) VALUES ('')
```

- L'instruction INSERT suivante est transmise à la source de données si vous divisez l'instruction en deux instructions distinctes :

```
INSERT INTO n1 SELECT * FROM n2
```

Vous devez impérativement utiliser deux instructions lorsque les tables cible ou source proviennent d'une source de données qui n'est pas compatible VARCHAR2, par exemple :

```
SELECT * FROM n2  
INSERT INTO n1 VALUES (:H0)
```

- Les fonctions qui sont imbriquées dans les expressions sont exécutées localement dans le serveur fédéré, par exemple :

```
SELECT LENGTH(TRIM(c1)) FROM n1
```
- Toutes les opérations de comparaison sont exécutées localement dans le serveur fédéré, car des résultats incohérents sont possibles lorsque les opérations de comparaison sont exécutées par la source de données.

Scénario 3

Le serveur fédéré et la source de données sont tous deux compatibles VARCHAR2. Ni le serveur fédéré ni la source de données n'autorise des chaînes vides ou n'utilise la sémantique de remplissage de blancs. Ainsi, toutes les opérations qui impliquent des chaînes vides sont transmises à la source de données, car la sémantique est préservée.

Séquence de classement

Si vous définissez l'option de serveur COLLATING_SEQUENCE sur 'Y', vous indiquez à la base de données fédérée que la séquence de classement de la source de données correspond à la séquence de classement du serveur fédéré. Ce paramètre permet à l'optimiseur d'étudier le transfert des traitements qui dépendent de l'ordre sur une source de données, qui permettrait d'améliorer les performances.

Si la séquence de classement de la source de données est différente de celle de la base de données fédérée et que vous avez défini l'option de serveur COLLATING_SEQUENCE sur 'Y', vous risquez de recevoir des résultats incorrects. Par exemple, si votre plan utilise des jointures par fusion, l'optimiseur peut transférer l'exécution des opérations de classement sur les sources de données. Si la séquence de classement de la source de données est différente, les résultats de la jointure risquent de ne pas être corrects dans l'ensemble. Définissez l'option de serveur COLLATING_SEQUENCE sur 'N', si vous n'êtes pas certain que la séquence de classement de la source de données est identique à celle de la base de données fédérée.

Sinon, vous pouvez configurer une base de données fédérée pour utiliser la même séquence de classement que celle utilisée par la source de données. Puis vous définissez l'option de serveur COLLATING_SEQUENCE sur 'Y'. Cette action autorise l'optimiseur à étudier le transfert de l'exécution des opérations qui dépendent de l'ordre sur les colonnes de caractères.

Pour déterminer si une source de données et la base de données fédérée ont la même séquence de classement, vous devez prendre en compte les facteurs suivants :

- Support de langue nationale
La séquence de classement est liée à la langue qui est prise en charge sur un serveur. Comparez les informations NLS de la base de données fédérée de votre système d'exploitation avec celles de la source de données.

- Classements sensibles à la langue
Vérifiez si la base de données fédérée ou la source de données utilise des collations sensibles à la langue. Si elles utilisent différents classements, vous ne devez pas définir COLLATING_SEQUENCE sur Y.
- Caractéristiques de la source de données
Certaines sources de données sont créées à l'aide des séquences de classement insensibles à la casse, qui peuvent renvoyer des résultats différents de la base de données fédérée dans des opérations qui dépendent de l'ordre.
- Personnalisation
Certaines sources de données fournissent plusieurs options aux séquences de classement ou permettent de personnaliser les séquences de classement.

Quand un fragment de requête d'un serveur fédéré nécessite un classement, l'endroit où s'effectue le classement dépend de plusieurs facteurs. Si la séquence de classement de la base de données fédérée est la même que celle de la source de données, le tri peut s'effectuer au niveau de la source de données ou du serveur fédéré. L'optimiseur de requêtes identifie le tri le plus efficace, local ou à distance, pour effectuer la requête.

En règle générale, les comparaisons numériques peuvent s'effectuer à n'importe quel endroit, même si la séquence de classement est différente. Vous risquez d'obtenir des résultats incorrects, toutefois, si la pondération des caractères nuls est différente entre la base de données fédérée et la source de données.

De même, pour des instructions de comparaison, vous devez être prudent si vous soumettez des instructions à une source de données insensible à la casse. Les pondérations attribuées aux caractères "I" et "i" dans une source de données insensible à la casse sont identiques. Par exemple, dans une source de données insensible à la casse avec une page de codes en anglais, **STEWART**, **SteWArT** et **stewart** seront tous considérés comme identiques. Par défaut, la base de données fédérée est sensible à la casse et pourrait attribuer différentes pondérations aux caractères.

Si les séquences de classement de la base de données fédérée et de la source de données sont différentes, le serveur fédéré extrait les données de la base de données fédérée, afin de pouvoir effectuer le classement localement. En effet, les utilisateurs s'attendent à voir les résultats de la requête triés conformément à la séquence de classement définie pour le serveur fédéré ; en classant les données localement, le serveur fédéré garantit des résultats conformes à cette attente.

Si votre requête contient un prédicat d'égalité sur la colonne de caractères, le transfert de l'exécution de cette partie de requête est possible même si les séquences de classement sont différentes (définies sur 'N'). Par exemple, le prédicat C1 = 'A' devrait extraire les mêmes valeurs quelle que soit la séquence de classement et par conséquent, son exécution peut être transférée sur une source de données qui contient une séquence de classement différente de celle du serveur fédéré. Toutefois, l'exécution de tels prédicats ne peut pas être transférée quand la séquence de classement de la source de données est insensible à la casse (COLLATING_SEQUENCE='I'). Quand une source de données est insensible à la casse, les résultats de C1= 'A' et C1 = 'a' sont identiques, ce qui est incohérent avec un environnement sensible à la casse tel que DB2 Database for Linux, UNIX et Windows.

Les administrateurs peuvent créer des bases de données fédérées avec une séquence de classement particulière qui correspond à la séquence de classement de

la source de données. Cette approche accélère les performances si toutes les sources de données utilisent la même séquence de classement ou si certaines ou toutes les fonctions de colonnes sont dirigées vers des sources de données qui utilisent la même séquence de classement. Par exemple, dans DB2 for z/OS, les tris définis par les clauses ORDER BY sont mis en oeuvre par une séquence de classement basée sur une page de codes EBCDIC. Si vous souhaitez utiliser le serveur fédéré pour extraire des données DB2 for z/OS classées conformément aux clauses ORDER BY, nous vous recommandons de configurer la base de données fédérée pour utiliser une séquence de classement prédéfinie basée sur la page de codes EBCDIC.

Si les séquences de classement de la base de données fédérée et de la source de données sont différentes, et que vous avez besoin d'afficher les données classées dans la séquence de la source de données, vous pouvez soumettre votre requête dans une session passe-système ou définir la requête dans une vue de source de données.

Options de serveur fédéré

Les options de serveur que vous avez définies détaillent les connaissances dont dispose le serveur fédéré sur la source de données éloignée.

Les facteurs répertoriés précédemment qui affectent les possibilités de transfert de l'exécution sur la base de données sont les caractéristiques des serveurs de la base et vous ne pouvez pas les modifier. En prenant attentivement en compte les options de serveur suivantes, vous pouvez améliorer les performances des requêtes :

- **COLLATING_SEQUENCE.** Si une source de données dispose d'une séquence de classement différente de celle de la base de données fédérée, toute opération sensible à l'ordre effectuée sur des valeurs de caractère ne peut pas être évaluée à distance sur la source de données. Par exemple, l'exécution des fonctions de colonne MAX sur une colonne de caractères de pseudonymes au niveau d'une source de données qui dispose d'une séquence de classement différente. Les résultats pouvant être différents si la fonction MAX est évaluée sur la source de données, la base de données fédérée va effectuer localement les opérations d'agrégat et la fonction MAX.
- **VARCHAR_NO_TRAILING_BLANKS.** Cette option s'adresse aux chaînes de caractères à longueur variable qui ne contiennent pas de blanc de fin. Certaines sources de données, telle que Oracle, n'appliquent pas de sémantique de comparaison espacée par un élément de remplissage comme le fait la base de données fédérée. Cette différence relative à l'élément de remplissage peut engendrer des résultats imprévus.

Par exemple :

```
'HELLO' = 'HELLO ' dans DB2  
'HELLO' <> 'HELLO ' dans Oracle!
```

Si les blancs de fin sont présents dans les colonnes VARCHAR d'une source de données Oracle, vous devez définir cette option sur N (valeur définie par défaut pour Oracle). Cette option agit sur les performances, car le serveur fédéré doit compenser la différence de sémantique, mais garantit un ensemble de résultats cohérent. La définition de cette valeur sur Y quand une colonne de source de données Oracle contient des blancs de fin peut engendrer des résultats incohérents.

Si vous êtes certain que toutes les colonnes VARCHAR et VARCHAR2 d'une source de données ne contiennent aucun blanc de fin, vous devez envisager de

définir cette option de serveur pour une source de données. Assurez-vous d'avoir bien étudié tous les objets qui peuvent avoir des pseudonymes, y compris les vues.

Recommandation : Définissez cette option colonne par colonne à l'aide de l'option de colonne `VARCHAR_NO_TRAILING_BLANKS`.

- `DB2_MAXIMAL_PUSHDOWN`. Cette option spécifie le principal critère dont se sert l'optimiseur de requêtes pour sélectionner un plan d'accès. L'optimiseur de requêtes sélectionne un plan d'accès en fonction du coût ou en fonction de l'exigence du client qui nécessite le traitement du plus grand nombre de requêtes possible par les sources de données éloignées. Avec l'option `DB2_MAXIMAL_PUSHDOWN` définie sur `Y`, la réduction du trafic réseau devient le critère prédominant de l'optimiseur de requête. L'optimiseur de requêtes utilise le plan d'accès qui effectue le moins d'"envois" aux sources de données. La définition de cette option de serveur sur `Y` impose au serveur fédéré d'utiliser un plan d'accès qui n'est pas forcément le plan le plus économique. L'utilisation d'un plan d'accès autre que le plan le plus économique risque de réduire les performances. Quand l'option de serveur `DB2_MAXIMAL_PUSHDOWN` est définie sur `Y`, l'exécution d'une requête ayant pour résultat un produit cartésien ne sera pas transférée sur les sources de données éloignées. Les requêtes qui aboutissent à un produit cartésien seront traitées par la base de données fédérée. L'option de serveur `DB2_MAXIMAL_PUSHDOWN` n'a pas besoin d'être définie sur `Y` pour que le serveur fédéré transfère l'exécution du traitement de la requête sur les sources de données éloignées. Quand cette option de serveur est définie sur `N` (valeur par défaut), l'optimiseur de requêtes va transférer l'exécution du traitement de la requête sur les sources de données. Toutefois, quand l'option est définie sur `N`, le premier critère sur lequel se base l'optimiseur de requêtes est le coût au lieu du trafic réseau.

«Caractéristiques de serveur ayant une incidence sur l'optimisation globale», à la page 275 décrit les options de serveur `COMM_RATE`, `CPU_RATIO` et `IO_RATIO` qui peuvent également avoir une incidence sur les performances des requêtes.

Facteurs de mappage de type et de fonction

Les mappages de types de données par défaut et les mappages de fonctions par défaut sont créés dans les encapsuleurs de source de données. Les mappages de types de données décrivent la relation entre le type de données de la source de données et celui du serveur fédéré. Vous pouvez personnaliser les mappages de types de données par défaut. Les mappages de fonctions décrivent la relation entre une fonction de source de données et une fonction équivalente du point de vue sémantique sur le serveur fédéré. Dans certains cas, la base de données fédérée compense pour les mappages de fonctions non pris en charge par une source de données.

Les mappages de types de données par défaut sont conçus de sorte qu'un espace de mémoire tampon suffisant soit affecté à chaque type de données de source de données pour éviter une troncature et un dépassement de la mémoire tampon d'exécution. Vous pouvez personnaliser le mappage des types d'une source de données spécifique ou d'un pseudonyme particulier pour des applications spécifiques et dans certains cas, pour améliorer les performances. Par exemple, les types `DATE` Oracle peuvent contenir à la fois une date et une portion horodatage et par conséquent, être mappés aux `HORODATAGES` DB2 par défaut. Si vous accédez à une colonne de date Oracle, et que vous savez qu'elle contient uniquement des parties de date (pas d'horodatage), vous pouvez utiliser l'instruction `ALTER NICKNAME` pour modifier le type de données local du

pseudonyme de `TIMESTAMP` à `DATE` lorsque l'option de serveur `date_compat` est désactivée. Lors d'une évaluation des prédicats strictement basée sur une date, tels que `SalesDate=DATE('2009-01-04')`, cette modification ignore l'utilisation d'une fonction `SCALAR` qui permet d'extraire la date des données de date et d'horodatage conservées dans la colonne, ce qui peut améliorer les performances.

La base de données fédérée compense pour les fonctions non prises en charge par une source de données. La compensation fonctionnelle implique généralement l'extraction des données nécessaires de la source de données et l'application de la fonction en local, ce qui a souvent un impact sur les performances. La compensation des fonctions survient dans plusieurs cas :

- Une fonction n'existe pas dans la source de données. Certaines des fonctions `SYSFUN` n'existent pas dans les sources de données `DB2 for z/OS` et requièrent donc une compensation locale.
- Il existe une fonction dans la source de données, mais les caractéristiques de l'opérande transgressent les restrictions de cette fonction (par exemple, l'opérateur de relation `NULL`). La plupart des sources de données le prennent en charge, mais pour d'autres, il existe des restrictions (par exemple, seul un nom de colonne est autorisé à gauche de l'opérateur `IS NULL`).
- Une fonction, si elle est évaluée à distance, peut renvoyer un résultat différent, comme c'est le cas de l'opérateur `'>'` (supérieur à). Pour ces sources de données qui possèdent des séquences de classement différentes, l'opérateur "supérieur à" peut renvoyer des résultats différents s'il est évalué en local par la base de données fédérée.

Caractéristiques de pseudonymes ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données

Parmi les caractéristiques de pseudonymes qui sont déterminants dans le transfert de l'exécution sur la base de données, vous trouvez le type de données locales d'une colonne de pseudonymes, les options de colonne fédérée et les tables de requêtes matérialisées.

Plusieurs facteurs propres aux pseudonymes sont déterminants dans le transfert des possibilités de transfert de l'exécution sur la base de données. Le type de données locales d'une colonne de pseudonyme peut déterminer le nombre de possibilités dans une séquence de jointure évaluée par l'optimiseur. Les pseudonymes peuvent être marqués d'une option de colonne pour indiquer celles qui ne contiennent aucun blanc de fin. En conséquence, le compilateur SQL a la possibilité de générer un format de prédicat plus efficace pour l'instruction SQL envoyée aux sources de données.

Type de données locales d'une colonne de pseudonyme

Assurez-vous que le type de données locales d'une colonne n'empêche pas l'évaluation d'un prédicat sur la source de données.

Les mappages de type de données par défaut permettent d'éviter tout dépassement de capacité potentiel. Toutefois, un prédicat de jointure entre deux colonnes de longueurs ou de types de données différents empêche l'optimiseur de prendre en compte la technique de jointure hachée. L'optimiseur pourra la prendre en compte uniquement si les longueurs et les types de données des colonnes de jointure correspondent exactement. Par exemple, les colonnes de la source de données Oracle conçues pour contenir uniquement des entiers sont souvent créées comme `NUMBER` dans la base de données Oracle, avec une valeur par défaut `NUMBER`

(38). Une colonne de pseudonyme pour ce type de données Oracle obtient le type de données locales FLOAT car l'intervalle d'un entier DB2 est à peu près égal à NUMBER (9). Dans ce cas, les jointures entre une colonne d'entiers DB2 et une colonne Oracle définie comme NUMBER (mais contenant uniquement des entiers) ne peuvent pas utiliser la technique de jointure hachée car la colonne Oracle est mappée comme type FLOAT. Toutefois, si le domaine de cette colonne NUMBER Oracle peut être adapté au type de données INTEGER DB2, vous pouvez modifier son type de données locales avec l'instruction ALTER NICKNAME. L'optimiseur peut ensuite étudier la technique de jointure hachée qui peut améliorer les performances.

Options de colonne fédérée

Vous pouvez définir des options de colonne fédérée qui permettront à l'optimiseur de requêtes de développer des plans d'accès.

Les options de colonne indiquent à l'encapsuleur de traiter les données d'une colonne d'une manière différente que celle utilisée habituellement. Les métadonnées permettent au compilateur SQL et à l'optimiseur de requêtes de développer de meilleurs plans d'accès aux données. La base de données fédérée traite les objets référencés par un pseudonyme comme s'il s'agissait d'une table. En conséquence, vous pouvez définir des options de colonne pour tous les objets de source de données pour lesquels vous créez un pseudonyme.

L'instruction ALTER NICKNAME permet d'ajouter ou de modifier des options de colonne des pseudonymes. Deux options de colonne existent :

- **NUMERIC_STRING.** Cette option de colonne s'applique aux colonnes de type caractères (CHAR et VARCHAR). Supposons qu'une source de données dispose d'une séquence de classement différente de celle de la base de données fédérée. Le serveur fédéré ne triera pas les colonnes qui contiennent des données de type caractères sur la source de données. Il va renvoyer les données à la base de données fédérée et effectuer un tri localement. Cependant, supposez que la colonne est un type de données caractères et contient uniquement des caractères numériques ('0','1',..., '9'). Vous pouvez l'indiquer en affectant la valeur 'Y' à l'option de colonne NUMERIC_STRING. Ainsi, l'optimiseur de requêtes peut choisir d'effectuer le tri sur la source de données car les caractères numériques, même représentés comme des chaînes de caractères, sont toujours triés de la même manière, quelle que soit la séquence de classement. Si le tri est effectué à distance, vous évitez la surcharge engendrée par le portage des données sur le serveur fédéré et l'opération de tri au niveau local.
- **VARCHAR_NO_TRAILING_BLANKS.** Contrairement à l'option de serveur du même nom, cette option de colonne permet d'identifier les colonnes Oracle spécifiques qui ne contiennent aucun blanc de fin. Le phase d'analyse du transfert de l'exécution du compilateur SQL va ensuite prendre en compte ces informations au moment de la vérification de toutes les opérations effectuées sur des colonnes définies avec ce paramètre. Basé sur le paramètre VARCHAR_NO_TRAILING_BLANKS, le compilateur SQL génère une sorte de prédicat différent mais sémantiquement équivalent, utilisé dans l'instruction SQL éloignée, envoyée à la source de données. La valeur 'Y' permet en principe d'utiliser les index éloignés (si disponibles) qui peuvent améliorer les performances des requêtes.

Caractéristiques de requête ayant une incidence sur les possibilités de transfert de l'exécution sur la base de données

Un opérateur SQL qui référence plusieurs sources de données est déterminant dans le transfert de l'exécution sur la base de données.

Une requête peut référencer un opérateur SQL qui implique des pseudonymes de plusieurs sources de données. Quand le serveur fédéré associe les résultats de deux sources de données référencées à l'aide d'un seul opérateur, l'opération soit s'effectuer sur le serveur fédéré. Par exemple, si l'opérateur est de type ensembliste, tel que UNION. L'opérateur ne peut pas être évalué directement sur une source de données éloignée.

Analyse de l'endroit où une requête est évaluée

Les informations détaillées relatives à l'optimiseur de requêtes sont conservées dans des tables Explain, séparées du plan d'accès actuel même. Ces informations permettent d'effectuer une analyse approfondie d'un plan d'accès. L'examen de l'opérateur SHIP d'un plan d'accès fédéré permet de déterminer les opérations SQL dont l'exécution a été transférée sur une source de données et celles qui ont été exécutées sur le serveur fédéré.

Les tables Explain, accessibles sur tous les systèmes d'exploitation pris en charge, contiennent des informations relatives aux instructions SQL dynamiques et statiques. En règle générale, les outils suivants permettent d'extraire les informations sur un plan d'accès des tables Explain :

- Outil de format de table Explain. L'outil db2exfmt permet de présenter les informations provenant de tables Explain dans un format prédéfini.
- Outils db2expln et dynexpln. Ces outils permettent de comprendre le plan d'accès qui a été sélectionné pour une instruction SQL particulière. Vous pouvez également vous servir de la fonction Explain intégrée dans le Centre de contrôle DB2 conjointement avec Visual Explain pour comprendre le plan d'accès qui a été sélectionné pour une instruction SQL particulière. Cette fonction Explain permet d'expliquer à la fois les instructions SQL dynamiques et statiques. A la différence des outils Explain, Visual Explain permet d'afficher les informations Explain dans un format graphique. Sinon, le niveau de détails fourni par les deux méthodes est équivalent. Pour une utilisation optimale de la sortie de db2expln et dynexpln, vous devez connaître :
 - Les différentes instructions SQL prises en charge et la terminologie liées à ces instructions (telles que les prédicats dans une instruction SELECT)
 - Le but d'un module (plan d'accès)
 - L'objectif et le contenu des tables du catalogue système
 - Les principales notions d'optimisation de l'application

Vous pouvez également accéder aux tables Explain à l'aide des instructions SQL. Cette action simplifie ainsi la manipulation du résultat, les comparaisons entre les différentes requêtes ou les comparaisons d'une même requête sur une période prolongée.

Analyse de l'endroit où une requête est évaluée avec l'option de serveur DB2_MAXIMAL_PUSHDOWN

Vous pouvez utiliser l'option de serveur DB2_MAXIMAL_PUSHDOWN avec les utilitaires Explain pour déterminer si l'exécution d'un opérateur particulier n'a pas été transférée vers la source de données en raison d'une décision économique de l'optimiseur ou car l'analyse du transfert de l'exécution sur la base de données a déterminé qu'elle n'était pas possible.

Procédure

Pour exécuter les outils Explain sur une requête à l'aide de l'option de serveur DB2_MAXIMAL_PUSHDOWN, procédez comme suit :

1. Affectez à l'option de serveur DB2_MAXIMAL_PUSHDOWN la valeur 'N'. Il s'agit de la valeur par défaut de cette option. L'analyse du transfert de l'exécution sur la base de données détermine les portions du code SQL dont l'exécution peut être transférée sur la base de données. L'optimiseur de requêtes génère ensuite tous les autres plans qui ne transgressent pas les critères définis par l'analyse du transfert de l'exécution sur la base de données. Il estime le coût de chacun et sélectionne le plan dont le coût prévu est le plus bas. Vous pouvez analyser les opérateurs dont l'exécution a été transférée vers la source de données en affichant les détails de l'opérateur SHIP approprié. Si vous pensez que l'exécution d'un opérateur aurait dû être transférée sur la base de données, mais qu'elle ne l'a pas été, passez à l'étape 2.
2. Affectez à l'option de serveur DB2_MAXIMAL_PUSHDOWN la valeur 'Y'. Réanalysez l'instruction SQL à l'aide des outils Explain. Le plan affiché dans les outils Explain montre toutes les opérations SQL dont l'exécution peut être transférée vers la source de données.
 - Si l'exécution de l'opérateur est transférée sur la base de données une fois que vous avez spécifié la valeur 'Y', cela signifie que l'optimiseur a déterminé qu'il était plus économique d'exécuter l'opérateur en local, plutôt qu'à distance. Si l'exécution de l'opérateur n'est pas transférée sur la base de données une fois que vous avez spécifié la valeur 'Y', il est probable que l'analyse du transfert de l'exécution sur la base de données n'a pas autorisé l'exécution à distance de l'opérateur.
 - Si l'optimiseur a déterminé qu'il n'était pas rentable de transférer l'exécution de l'opérateur sur la base de données, vérifiez si les statistiques du pseudonyme sont précises. Si l'analyse a choisi de ne pas transférer l'exécution de l'opérateur sur la base de données, vérifiez les options du serveur, les mappages de types de données et les mappages de fonctions.

Présentation des décisions d'évaluation de plan d'accès

Les rubriques de cette section présentent la liste des principales questions sur l'analyse des plans d'accès, et les points que vous pouvez étudier pour augmenter les possibilités de transfert de l'exécution sur la base de données.

Pourquoi ce prédicat n'est-il pas évalué à distance ?

Cette question est soulevée quand un prédicat est très sélectif et, par conséquent, peut être utilisé pour filtrer les lignes et réduire le trafic réseau. L'évaluation du prédicat à distance a également une incidence sur la possibilité ou non d'évaluer à distance une jointure entre deux tables d'une même source de données.

Les points à étudier sont les suivants :

- Options de serveur. Comment les paramètres des options de serveur `COLLATING_SEQUENCE` et `VARCHAR_NO_TRAILING_BLANKS` déterminent-ils l'endroit d'évaluation du prédicat ?
- Prédicats de sous-requête. Ce prédicat contient-il une sous-requête appartenant à une autre source de données ? Ce Prédicat contient-il une sous-requête qui implique un opérateur SQL non pris en charge par cette source de données ? Les sources de données ne prennent pas toutes en charge les opérateurs ensemblistes d'un prédicat.
- Fonctions de prédicat. Ce prédicat contient-il une fonction qui ne peut pas être évaluée par cette source de données éloignée ? Les opérateurs de relation sont classés comme des fonctions.
- Conditions de liaison d'un prédicat. Ce prédicat, si évalué à distance, nécessite-t-il la liaison de certaines valeurs ? Si tel est le cas, cette action transgresserait-elle les restrictions SQL de la source de données ?
- Optimisation globale. L'optimiseur a décidé que le traitement local est plus économique.

Pourquoi l'opérateur GROUP BY n'est-il pas évalué à distance ?

Vous pouvez examiner plusieurs points afin de déterminer les raisons pour lesquelles un opérateur GROUP BY n'est pas évalué à distance.

Les points que vous pouvez contrôler incluent :

- L'entrée de l'opérateur GROUP BY est-elle évaluée à distance ? En cas de réponse négative, étudiez l'entrée.
- La source de données a-t-elle des restrictions sur cet opérateur ? Voici quelques exemples :
 - Nombre maximal d'éléments GROUP BY
 - Nombre maximal d'octets pour des éléments GROUP BY associés
 - Spécification de colonne uniquement sur la liste GROUP BY
- La source de données prend-elle en charge cet opérateur SQL ?
- Optimisation globale. L'optimiseur a décidé que le traitement local est plus économique.

Pourquoi l'opérateur SET n'est-il pas évalué à distance ?

Vous pouvez vérifier les opérandes ainsi que les restrictions de la source de données pour déterminer les raisons pour lesquelles l'opérateur SET n'est pas évalué à distance.

Considérations :

- Ses deux opérandes sont-elles intégralement évaluées au niveau de la même source de données éloignée ? Si la réponse est négative alors qu'elle devrait être positive, étudiez chaque opérande.
- La source de données a-t-elle des restrictions sur cet opérateur SET ? Par exemple, les objets volumineux ou les champs étendus sont-ils des entrées valides pour cet opérateur SET spécifique ?

Pourquoi l'opération ORDER BY n'est-elle pas évaluée à distance ?

Vous pouvez vérifier l'entrée de l'opération, le contenu de la clause, ainsi que les restrictions de la source de données pour déterminer les raisons pour lesquelles l'opérateur ORDER BY n'est pas évalué à distance.

Considérations :

- Options de serveur. Comment les paramètres des options de serveur COLLATING_SEQUENCE et VARCHAR_NO_TRAILING_BLANKS déterminent-ils l'endroit d'évaluation du prédicat ?
- L'entrée de l'opération ORDER BY est-elle évaluée à distance ? En cas de réponse négative, étudiez l'entrée.
- La clause ORDER BY contient-elle une expression de type caractère ? Dans l'affirmative, la source de données éloignée a-t-elle une séquence de classement différente de celle du serveur fédéré ?
- La source de données a-t-elle des restrictions sur cet opérateur ? Par exemple, existe-t-il un nombre maximal d'éléments ORDER BY ? La source de données limite-t-elle les spécifications de colonne de la liste ORDER BY ?

Pourquoi une instruction INSERT éloignée avec une instruction FULLSELECT n'est-elle pas intégralement évaluée à distance ?

Vous pouvez vérifier plusieurs éléments de la sous-requête pour déterminer pourquoi une instruction INSERT éloignée avec une instruction FULLSELECT n'est pas intégralement évaluée à distance.

Considérations :

- La sous-requête peut-elle être intégralement évaluée au niveau de la source de données éloignée ? Si la réponse est non, examinez la sous-requête.
- La sous-requête contient-elle un opérateur ensembliste ? Dans l'affirmative, cette source de données prend-elle en charge les opérateurs ensemblistes comme entrée d'une instruction INSERT ?
- La sous-requête référence-t-elle la table cible ? Si la réponse est oui, cette source de données autorise-t-elle cette syntaxe ?

Pourquoi une instruction INSERT éloignée avec une clause VALUES n'est-elle pas intégralement évaluée à distance ?

Vous pouvez vérifier la clause VALUES et l'expression afin de comprendre pourquoi une instruction INSERT éloignée avec une clause VALUES n'est pas intégralement évaluée à distance.

Considérations :

- La clause VALUES peut-elle être intégralement évaluée au niveau de la source de données éloignée ? En d'autres termes, une expression contient-elle une fonction non prise en charge par la source de données éloignée ?
- L'expression implique-t-elle une sous-requête scalaire ? Cette syntaxe est-elle prise en charge ?
- L'expression référence-t-elle la table cible ? Cette syntaxe est-elle prise en charge ?

Pourquoi une instruction UPDATE recherchée à distance n'est-elle pas intégralement évaluée à distance ?

Vous pouvez vérifier les éléments de la clause SET et la condition de recherche pour déterminer les raisons pour lesquelles une instruction UPDATE recherchée à distance n'est pas intégralement évaluée à distance.

Considérations :

- La clause SET peut-elle être intégralement évaluée au niveau de la source de données éloignée ? En d'autres termes, une expression de mise à jour contient-elle une fonction non prise en charge par la source de données éloignée ?
- La clause SET implique-t-elle une sous-requête scalaire ? La source de données permet-elle cette syntaxe ?
- La condition de recherche peut-elle être intégralement évaluée au niveau de la source de données éloignée ? Si la réponse est non, examinez la condition de recherche.
- La condition de recherche ou la clause SET fait-elle référence à la table cible ? La source de données permet-elle cette syntaxe ?
- La condition de recherche ou la clause SET fait-elle référence à la table cible avec une corrélation ? La source de données permet-elle cette syntaxe ?

Pourquoi une instruction UPDATE de position n'est-elle pas intégralement évaluée ?

Cela se produit quand la base de données fédérée choisit d'évaluer l'expression de mise à jour localement avant d'envoyer l'instruction UPDATE à la source de données. Cette approche ne devrait pas avoir d'incidence considérable sur les performances.

Considérations :

- La clause SET peut-elle être intégralement évaluée au niveau de la source de données éloignée ? En d'autres termes, une expression de mise à jour contient-elle une fonction non prise en charge par la source de données éloignée ?
- La clause SET implique-t-elle une sous-requête scalaire ? La source de données permet-elle cette syntaxe ?

Pourquoi une instruction DELETE de recherche, éloignée, n'est-elle pas intégralement évaluée à distance ?

Vous pouvez vérifier les éléments de la condition de recherche afin de déterminer pourquoi une instruction DELETE de recherche, éloignée, n'est pas intégralement évaluée à distance.

Considérations :

- La condition de recherche peut-elle être intégralement évaluée au niveau de la source de données éloignée ? Si la réponse est non, examinez la condition de recherche.
- La condition de recherche référence-t-elle la table cible ? La source de données permet-elle cette syntaxe ?
- La condition de recherche référence-t-elle la table cible avec corrélation ? La source de données permet-elle cette syntaxe ?

Mises à niveau et personnalisation des sources de données

Quand les sources de données sont mises à niveau ou personnalisées, vous devez actualiser les informations du catalogue global.

Le compilateur SQL s'appuie sur les informations qui sont stockées dans le catalogue global pour lui fournir les fonctionnalités SQL des sources de données. Ces informations nécessitent des mises à jour périodiques. Les fonctionnalités SQL des sources de données risquent d'être modifiées dans les nouvelles versions des sources de données. Quand les sources de données sont mises à jour ou personnalisées, vous devez actualiser les informations du catalogue global afin que le compilateur SQL utilise les informations les plus récentes.

Les instructions SQL DDL, telles que CREATE FUNCTION MAPPING et ALTER SERVER, permettent de mettre à jour le catalogue.

Transfert de l'exécution sur la base de données des prédicats avec le modèle de fonction

Dans un système fédéré, chaque source de données éloignée est dotée de ses propres fonctions. La plupart de ces fonctions ont des fonctions DB2 sémantiquement équivalentes ainsi que des mappages de fonctions associés par défaut. Toutefois, certaines fonctions de source éloignée risquent de ne pas avoir de fonctions équivalentes sur le serveur fédéré. Par conséquent, seule la source de données éloignée peut exécuter ces fonctions. Pour écrire des requêtes qui utilisent ces fonctions, vous devez créer un modèle de fonction sur le serveur fédéré.

Un modèle de fonction agit comme une description locale de la fonction éloignée. Vous créez un modèle de fonction avec l'instruction CREATE FUNCTION à l'aide de la clause AS TEMPLATE. Aucun code exécutable n'est associé au modèle de fonction sur le serveur fédéré. Quand le modèle est défini, vous pouvez l'utiliser pour créer un mappage de fonctions, qui associe le modèle de fonction à son homologue éloigné. Il est ensuite possible de se référer au modèle de fonction dans les instructions SQL qui sont émises sur le serveur fédéré et d'évaluer la fonction sur la source de données.

L'optimiseur de requête prend des décisions en fonction des coûts pour déterminer l'endroit où peut être évalué un prédicat. Le cas échéant, l'optimiseur génère un plan pour évaluer un prédicat avec un modèle de fonction sur le serveur éloigné correspondant. Dans la plupart des cas, l'optimiseur risque de ne pas pouvoir générer de plan pour évaluer le modèle de fonction sur la source de données. Quand cela se produit, vous risquez de recevoir une erreur SQL0142N avec le message d'erreur suivant :

L'instruction SQL n'est pas prise en charge.

Pour éviter cette erreur, vous pouvez récrire la requête pour autoriser le transfert de l'exécution sur la base de données tout en conservant la sémantique de la requête originale.

Pour permettre le transfert de l'exécution d'un modèle de fonction, il doit être défini avec les clauses DETERMINISTIC et NO EXTERNAL ACTION.

Chapitre 24. Parallélisme avec des requêtes qui font référence aux pseudonymes

Les requêtes qui contiennent des pseudonymes peuvent participer à trois types de parallélisme intrarequête.

Les trois types de parallélisme intrarequête sont les suivants :

- Parallélisme de requête intrapartition sur une partition unique, configurations multiprocesseurs
- Parallélisme de requête interpartition dans des configurations à partitions multiples
- Parallélisme de requête mixte qui consiste en un parallélisme à la fois intrapartition et interpartition où chaque partition s'exécute sur un ordinateur SMP

Parallélisme intrapartition avec des requêtes qui font référence aux pseudonymes

Le parallélisme intrapartition se rapporte au processus de division d'une requête en plusieurs parties simultanées exécutées en parallèle par plusieurs processus sur une seule partition de base de données.

Dans des requêtes fédérées, la partie de requête qui implique des données locales peut s'exécuter en parallèle tandis que la partie qui implique des pseudonymes s'exécute de manière séquentielle, à l'aide d'un seul processus agent.

Quand plusieurs processeurs peuvent fonctionner sur les parties locales de la requête, les performances des requêtes qui se réfèrent aux pseudonymes et aux tables locales peuvent être améliorées.

Le paramètre de configuration de base de données DFT_DEGREE et le registre spécial CURRENT DEGREE contrôlent le degré de parallélisme intrapartition.

Activation du parallélisme intrapartition avec les requêtes qui référence des pseudonymes

Pour les requêtes qui font référence à des tables locales et à des pseudonymes dans un environnement multiprocesseur, vous pouvez activer le parallélisme intrapartition. Le serveur fédéré peut alors traiter les tables locales en parallèle.

Restrictions

Le système fédéré ne peut traiter que la portion d'une requête qui fait référence à des tables locales en parallèle. La partition coordinatrice traite toutes les opérations sur la portion éloignée d'une requête en série.

Procédure

Pour activer le parallélisme intrapartition, procédez comme suit :

1. Affectez la valeur YES au paramètre de configuration de base de données INTRA_PARALLEL.

2. Affectez au paramètre de configuration de base de données MAX_QUERYDEGREE une valeur supérieure à 1.
3. Affectez au paramètre de configuration de base de données DFT_DEGREE une valeur supérieure à 1 ou définissez le registre spécial CURRENT DEGREE. Si vous avez affecté au paramètre DFT_DEGREE la valeur ANY, le niveau par défaut du parallélisme intrapartition est égal au nombre de processeurs de l'ordinateur.

Parallélisme intrapartition avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès

L'utilitaire DB2 Explain permet d'afficher le plan d'accès utilisé par l'optimiseur lors du traitement des requêtes. Les exemples suivants montrent comment l'optimiseur accède aux données de pseudonymes dans un environnement qui utilise le parallélisme intrapartition.

Exemple 1 : sans prise en charge du parallélisme

Dans cet exemple, le serveur fédéré traite la jointure de la table locale (ORDERS) et du pseudonyme (ITEMS) en série. Aucun parallélisme intrapartition n'est utilisé.

```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

```

      RETURN
      ( 1)
      |
      HSJOIN
      ( 2)
    /-----\
    TBSCAN SHIP
    ( 3) ( 4)
    |           |
    TABLE: NEWTON NICKNM: NEWTON
    ORDERS ITEMS
```

Exemple 2 : avec prise en charge du parallélisme

Dans cet exemple de jointure, la requête peut être exécutée plus rapidement si la table locale est lue en parallèle avant sa jointure en série avec le pseudonyme. L'opérateur LTQ indique à quel endroit le parallélisme est introduit dans le plan.

```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

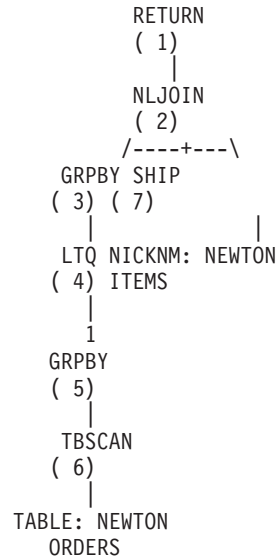
```

      RETURN
      ( 1)
      |
      HSJOIN
      ( 2)
    /-----\
    LTQ SHIP
    ( 3) ( 5)
    |           |
    TBSCAN NICKNM: NEWTON
    ( 4) ITEMS
    |
    TABLE: NEWTON
    ORDERS
```

Exemple 3 : parallélisme intrapartition avec agrégation

Dans cet exemple, la base de données regroupe les données de la table locale en parallèle dans la partition, ce qui améliore l'exécution de l'agrégation. La jointure de la table locale avec le pseudonyme est effectuée en série.

```
SELECT *
FROM ITEMS A
WHERE ID =
  (SELECT MAX(ID)
   FROM ORDERS
   WHERE NUMBER = 10)
```



Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes

Le parallélisme interpartition se rapporte au processus de division d'une requête unique en plusieurs parties exécutées en parallèle sur différentes partitions d'une base de données partitionnées.

Dans les requêtes qui font référence à des données locales et distantes, le serveur fédéré peut distribuer les données distantes à chacune des partitions locales. Les figure 10 et figure 11, à la page 255 présentent la notion de parallélisme interpartition qui implique des sources de données locales et distantes.

La figure 10 montre comment ce type de requête est traité sans parallélisme interpartition. Les données de pseudonymes éloignées et les données partitionnées locales sont traitées de manière séquentielle sur la partition coordinatrice. Cette technique n'utilise pas la puissance parallèle des partitions de la base de données car le traitement est effectué en grande partie sur une partition unique. Si les volumes de données sont très importants, cette technique risque d'engendrer de longues sessions de requêtes.

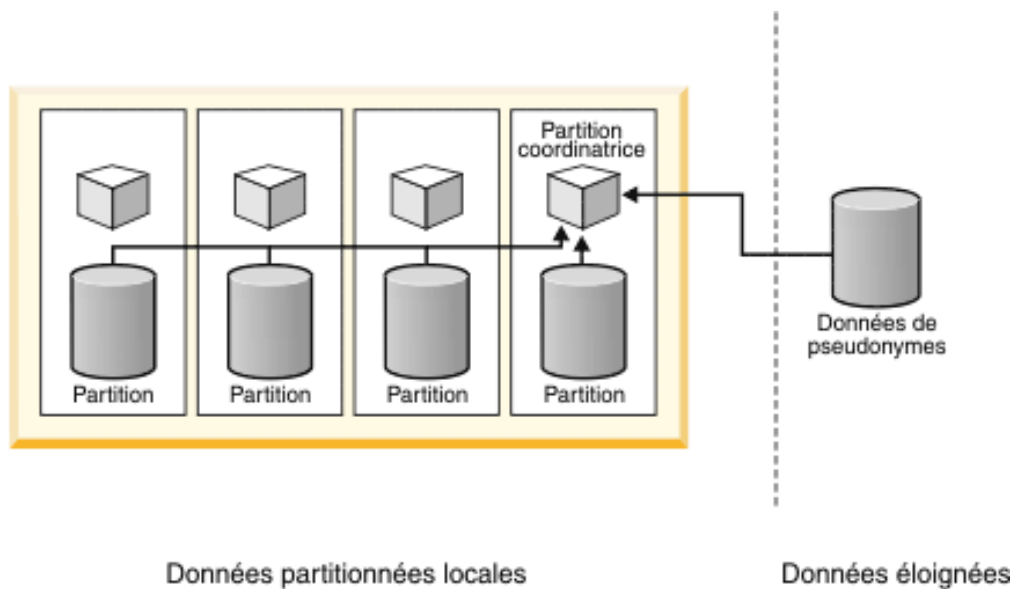


Figure 10. Requête sans parallélisme interpartition de sources de données locales et éloignées

La figure 11 décrit le traitement quand l'optimiseur distribue les données de pseudonymes sur les partitions. La partition coordinatrice extrait les données de pseudonymes et les distribue aux partitions de base de données pour un traitement parallèle. Quand le traitement parallèle est terminé, les résultats sont renvoyés à la partition coordinatrice pour un traitement final avant d'être renvoyés à l'application.

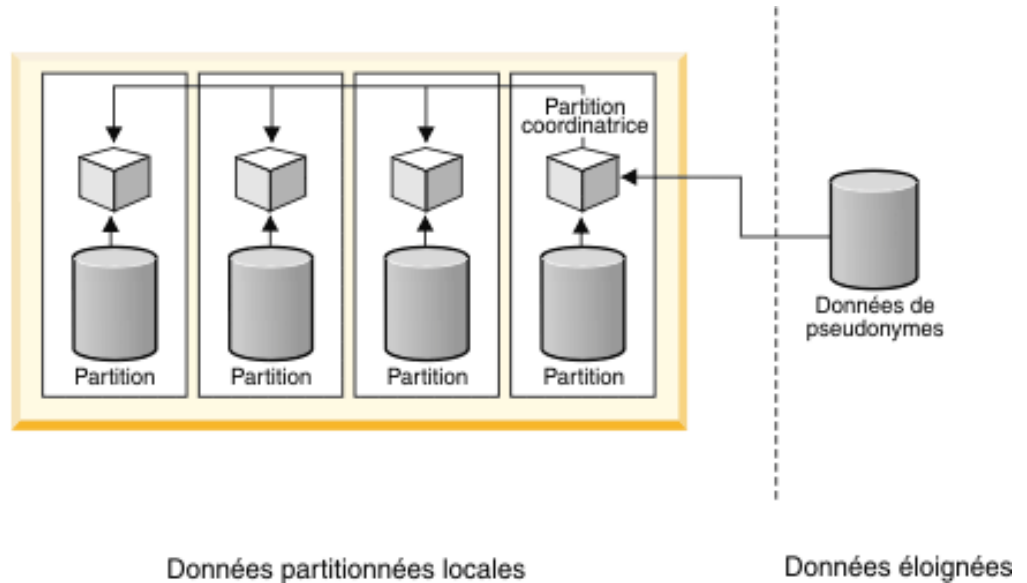


Figure 11. Requête avec parallélisme interpartition de sources de données locales et éloignées

Les figure 12 et figure 13, à la page 257 présentent la notion de parallélisme interpartition qui implique uniquement des sources de données éloignées.

La figure 12 montre le traitement en série des données de pseudonymes éloignées sur la partition coordinatrice. La partition coordinatrice, qui agit également en tant que serveur fédéré, extrait les données de pseudonymes et les traite de manière séquentielle.

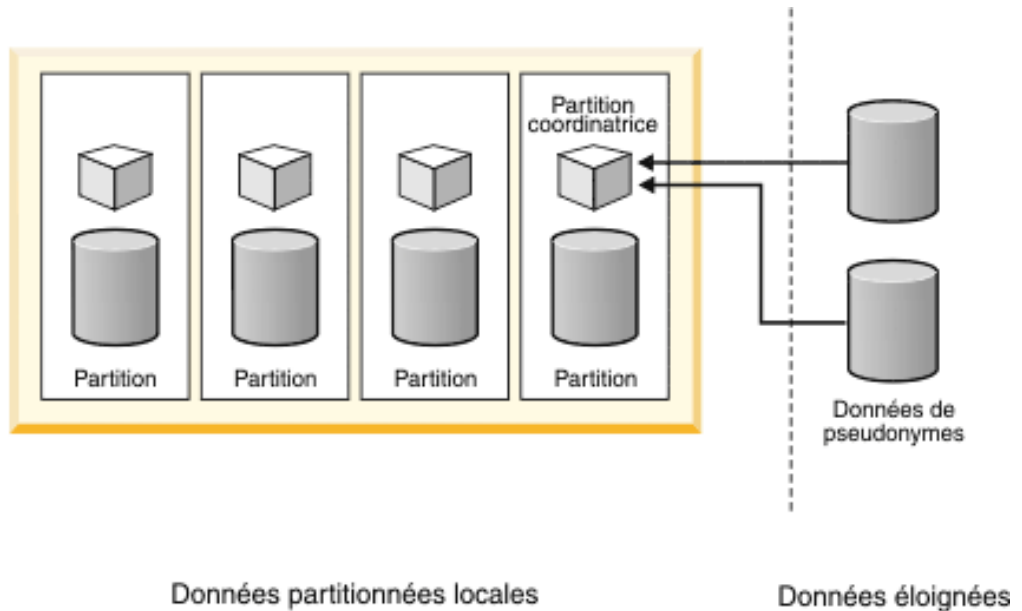


Figure 12. Requête sans parallélisme interpartition qui référence uniquement les sources de données éloignées.

La figure 13, à la page 257 montre la partition coordinatrice distribuant les données sur un groupe de partitions calculé. Les groupes de partition informatiques permettent à l'optimiseur de générer des plans d'accès qui distribuent des données de pseudonymes sur les partitions d'un serveur de base de données partitionnée pour le traitement en parallèle.

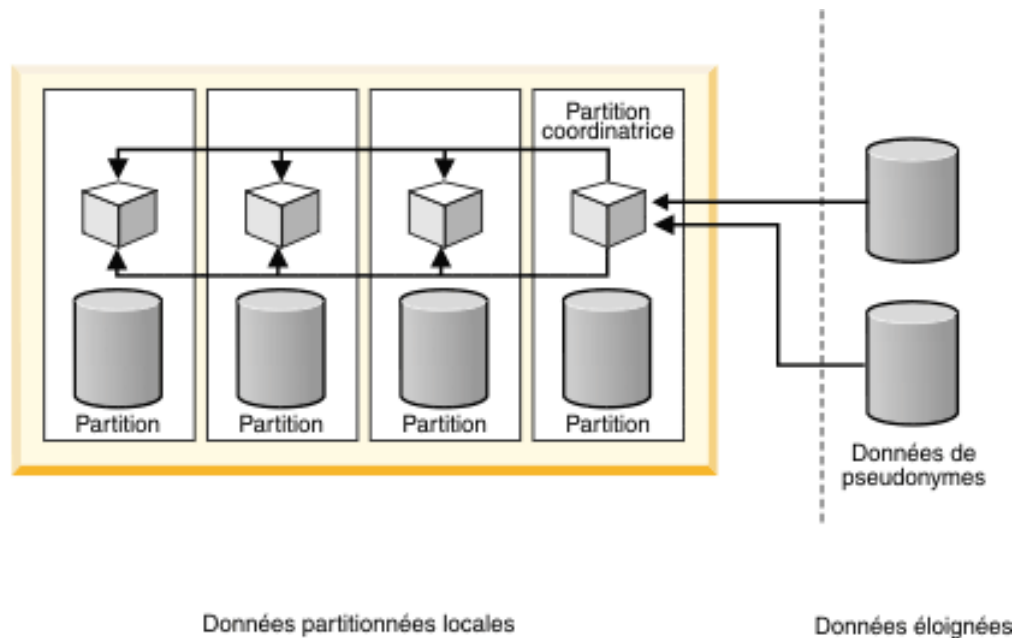


Figure 13. Requête avec parallélisme interpartition qui référence uniquement les sources de données éloignées.

Quel que soit le plan choisi par l'optimiseur, l'accès aux données de pseudonymes s'effectue toujours de manière séquentielle depuis la partition coordinatrice.

Activation du parallélisme interpartition avec les requêtes qui font référence à des pseudonymes

Vous pouvez configurer un serveur fédéré partitionné pour que les requêtes utilisant des pseudonymes puissent éventuellement être exécutées en parallèle sur plusieurs partitions. L'exécution en parallèle peut considérablement réduire la durée des requêtes fédérées dans un environnement partitionné.

Restrictions

Seules les portions d'une requête faisant référence à des pseudonymes qui utilisent des encapsuleurs isolés peuvent être exécutées en parallèle. Les portions d'une requête faisant référence à des pseudonymes qui utilisent des encapsuleurs sécurisés ne peuvent pas être exécutées en parallèle.

A propos de cette tâche

Dans un environnement de base de données partitionnée, les requêtes fédérées peuvent être exécutées en parallèle à certaines conditions :

- Elles utilisent une combinaison de pseudonymes définis à l'aide d'un encapsuleur isolé et de tables partitionnées locales
- Elles utilisent des pseudonymes définis à l'aide d'un encapsuleur isolé et un groupe de partitions calculé est défini.

Vous n'avez pas besoin de définir des paramètres de base de données ou des paramètres de configuration de base de données dans un environnement partitionné pour que le parallélisme interpartition soit disponible pour les requêtes fédérées.

Procédure

Pour activer le parallélisme interpartition, procédez comme suit :

1. Exécutez l'instruction CREATE WRAPPER ou ALTER WRAPPER en affectant la valeur Y à l'option DB2_FENCED.
2. **Facultatif** : Configurez un groupe de partitions calculé, si vous avez l'intention d'activer le parallélisme pour les portions de requête qui n'utilisent que des pseudonymes. Si des requêtes utilisent une combinaison de pseudonymes et de tables partitionnées locales, vous n'avez pas besoin de configurer de groupe de partitions calculé.

Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès

L'utilitaire DB2 Explain permet d'afficher le plan d'accès utilisé par l'optimiseur lors du traitement des requêtes. Les exemples suivants montrent comment l'optimiseur accède aux données de pseudonymes dans un environnement qui utilise le parallélisme interpartition.

Exemple 1 : mode sécurisé

Dans cet exemple, le pseudonyme utilise un encapsuleur sécurisé. La base de données effectue la jointure entre la table locale et le pseudonyme en série sur la partition coordinatrice. La base de données apporte les données locales, qui sont distribuées sur deux partitions, à la partition coordinatrice. Le serveur fédéré joint ensuite les données locales aux données de pseudonymes. La base de données joint en série les pseudonymes définis à l'aide d'un encapsuleur sécurisé sur la partition coordinatrice. La base de données ne peut pas distribuer les données sur plusieurs partitions pour créer une jointure parallèle.

```
SELECT *  
FROM ORDERS A, ITEMS B  
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

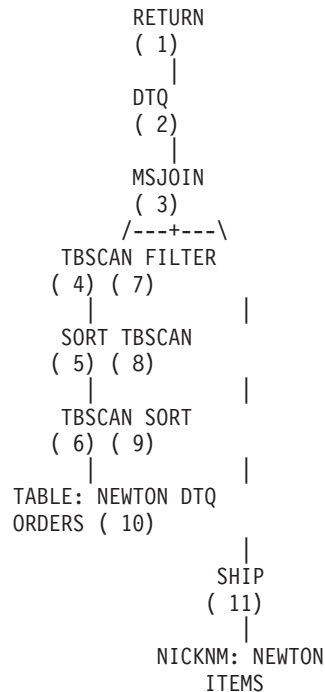
```
          RETURN  
          ( 1)  
          |  
        HSJOIN  
          ( 2)  
    /-----+-----\  
    DTQ SHIP  
    ( 3) ( 5)  
    |  
    TBSCAN NICKNM: NEWTON  
    ( 4) ITEMS  
    |  
    TABLE: NEWTON  
    ORDERS
```

Exemple 2 : mode isolé

Dans cet exemple, le pseudonyme utilise un encapsuleur isolé. Le serveur fédéré distribue les données de pseudonymes aux autres partitions et effectue la jointure avec les données locales en parallèle. L'opérateur DTQ (file d'attente de table distribuée) placé avant l'opérateur SHIP indique que les données de pseudonymes sont distribuées aux partitions locales à l'aide d'un partitionnement par adressage calculé pour obtenir une jointure colocalisée en parallèle. Dans une jointure colocalisée en parallèle, les données de pseudonymes sont distribuées aux

partitions locales de telle manière que les données locales et pseudonyme correspondant pour la jointure se trouvent toujours sur la même partition.

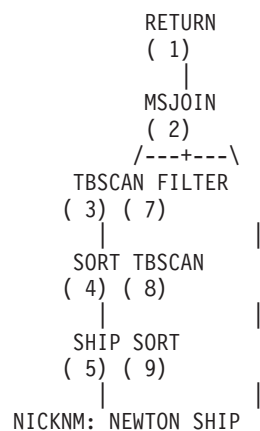
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Exemple 3 : mode isolé sans groupe de partitions calculé

Dans cet exemple, les deux pseudonymes utilisent un encapsuleur isolé et aucun groupe de partitions calculé n'est défini. Le serveur fédéré effectue la jointure sur la partition coordinatrice. Il ne distribue pas les données aux autres partitions en vue de leur traitement. L'absence d'opérateurs TQ au dessus des opérateurs SHIP indique que les données de pseudonymes ne sont pas distribuées sur les partitions.

```
SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID1 = B.ID1
```



```

LOCATIONS ( 10)
      |
      NICKNM: NEWTON
      ITEMS

```

Exemple 4 : mode isolé avec groupe de partitions calculé

Dans cet exemple, les pseudonymes utilisent des encapsuleurs isolés et un groupe de partitions calculé est défini. Dans ce cas, l'optimiseur sélectionne un plan qui distribue les données de la partition coordinatrice aux autres partitions du groupe de partitions calculé. Les opérateurs DTQ placés avant les deux pseudonymes partitionnent par adressage calculé les données éloignées entrantes de sorte que les clés de jonction correspondantes se trouvent sur la même partition du groupe de partitions calculé. La jointure est effectuée en parallèle sur chaque partition et les résultats sont ensuite collectés sur la partition coordinatrice.

```

SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID = B.ID

```

```

      RETURN
      ( 1)
      |
      DTQ
      ( 2)
      |
      MSJOIN
      ( 3)
      /---+---\
      TBSCAN FILTER
      ( 4) ( 9)
      |
      SORT TBSCAN
      ( 5) ( 10)
      |
      DTQ SORT
      ( 6) ( 11)
      |
      SHIP DTQ
      ( 7) ( 12)
      |
      NICKNM: NEWTON SHIP
      LOCATIONS ( 13)
      |
      NICKNM: NEWTON
      ITEMS

```

Groupes de partitions calculés

Un groupe de partitions calculé définit un ensemble de partitions de base de données pouvant être utilisées par l'optimiseur pour redistribuer des données de pseudonyme de manière dynamique. Un groupe de partitions calculé concerne les parties de requêtes qui impliquent uniquement les pseudonymes.

La partition coordinatrice extrait les données de pseudonymes par ordre puis les redistribue sur les partitions du groupe de partitions calculé, à l'endroit où le traitement parallèle s'effectue. L'utilisation de groupes de partitions informatiques par l'optimiseur améliore souvent les performances, particulièrement lorsque de nombreuses données de pseudonymes sont impliquées ou lorsque les requêtes sont complexes.

Un groupe de partitions calculé est un groupe de partitions de base de données, autre que IBM CATNODEGROUP, qui est spécifié dans le catalogue système, SYSCAT.DBPARTITIONGROUPS.

La variable de registre DB2_COMPPARTITIONGROUP permet de spécifier le groupe de partitions calculé.

Définition d'un groupe de partitions calculé

La définition d'un groupe de partitions calculé permet à l'optimiseur d'utiliser un plan qui distribue les données de pseudonymes aux partitions du groupe de partitions calculé. Vous définissez un groupe de partitions calculé pour activer le parallélisme interpartition des requêtes ou portions de requête qui ne font référence qu'à des pseudonymes.

Avant de commencer

Tous les groupes de partition utilisés pour représenter le groupe de partitions calculé sur l'ensemble des bases de données de l'instance doivent porter le même nom. Vous pouvez définir ces groupes de partition différemment dans chaque base de données, mais ils doivent tous porter le même nom. Par exemple, les trois bases de données DB1, DB2 et DB3 définissent un groupe de partitions calculé contenant des noeuds différents :

- DB1 : CPG contient les noeuds 1, 2, 3 et 4
- DB2 : CPG contient les noeuds 49, 50 et 53
- DB3 : CPG contient les noeuds 78 et 96

Vous pouvez affecter le nom CPG à la variable db2set. Le nom CPG est commun à l'ensemble des bases de données, mais le contenu du groupe de partitions calculé diffère pour chacune d'elles.

Restrictions

L'optimiseur n'utilise les groupes de partition informatiques que pour les portions d'une requête qui font référence à des pseudonymes, mais pas à des données locales.

Procédure

Pour définir un groupe de partitions calculé, exécutez la commande ci-après en ligne de commande DB2.

```
db2set  
DB2_COMPPARTITIONGROUP=nom_groupepartition
```

nom_groupepartition représente le nom du groupe de partitions à définir comme groupe de partitions calculé. Ce groupe de partitions doit être déjà défini.

L'exemple ci-après illustre comment définir le groupe de partitions calculé, FINANCE3, à l'aide de la variable de registre DB2_COMPPARTITIONGROUP.

```
db2set DB2_COMPPARTITIONGROUP=FINANCE3
```

Parallélisme interpartition avec des requêtes qui font référence aux pseudonymes - Attentes en matière de rendement

Pour les requêtes qui font référence à une combinaison de tables partitionnées locales et de pseudonymes, l'optimiseur peut choisir un plan d'exécution qui redistribue les données de pseudonymes sur les partitions appropriées.

Les plans de redistribution peuvent permettre une exécution plus rapide des requêtes si la quantité de données de pseudonymes de la jointure est inférieure à la quantité de données locales partitionnées. Si la quantité de données de pseudonymes de la jointure est bien supérieure à celle des données locales, il est peu probable qu'un plan parallèle avec redistribution des données de pseudonymes soit utilisé. Si l'optimiseur ne choisit pas de plan parallèle, le serveur fédéré effectue les jointures en série entre les pseudonymes et les tables locales sur la partition coordinatrice.

Pour les jointures entre deux pseudonymes, un plan d'exécution qui distribue les données entre toutes les partitions d'un groupe de partitions calculé peut être intéressant lorsque la quantité de données est importante. L'avantage de traiter la jointure de grande taille en parallèle compense le coût supplémentaire de la redistribution des données sur plusieurs partitions. Si la quantité de données de pseudonymes est relativement faible, le coût de la jointure n'est pas assez élevé pour compenser le coût supplémentaire de la redistribution des données sur les partitions. En général, l'optimiseur choisit des plans de groupe de partitions calculé si les pseudonymes impliqués sont de grande taille ; dans le cas contraire, le serveur fédéré joint les pseudonymes en série sur la partition coordinatrice.

Parallélisme mixte avec des requêtes qui font référence aux pseudonymes

Pour les requêtes qui contiennent des pseudonymes et des tables en local dans un environnement partitionné, l'optimiseur peut utiliser à la fois le parallélisme intrapartition et interpartition. Dans un environnement partitionné, le parallélisme interpartition est une option pour l'optimiseur. Le parallélisme intrapartition est une option, s'il a été activé dans la configuration de la base de données ou dans celle du gestionnaire de bases de données.

Dans le cadre du parallélisme interpartition, le serveur fédéré peut distribuer les données éloignées sur les partitions et les traiter en parallèle avec chaque partition.

Dans le cadre du parallélisme intrapartition, plusieurs processus sous-agents d'une partition permettent de traiter les données locales en parallèle.

Activation du parallélisme mixte avec les requêtes qui font référence à des pseudonymes

Vous pouvez améliorer les performances des requêtes qui font référence à des données locales et éloignées par l'intermédiaire du parallélisme intrapartition et interpartition.

Procédure

Pour activer le parallélisme interpartition sur un serveur fédéré partitionné, procédez comme suit :

1. Exécutez l'instruction CREATE WRAPPER ou ALTER WRAPPER en affectant la valeur Y à l'option DB2_FENCED.
2. **Facultatif** : Configurez un groupe de partitions informatique pour activer le parallélisme pour les jointures de pseudonymes uniquement.

Pour activer le parallélisme intrapartition sur un serveur fédéré, procédez comme suit :

1. Affectez au paramètre de configuration de base de données MAX_QUERYDEGREE une valeur supérieure à 1.
2. Affectez au paramètre de configuration de base de données DFT_DEGREE une valeur supérieure à 1 ou définissez le registre spécial CURRENT DEGREE. Si vous avez affecté au paramètre DFT_DEGREE la valeur ANY, le niveau par défaut du parallélisme intrapartition est égal au nombre de processeurs SMP de l'ordinateur.

Parallélisme mixte avec des requêtes qui font référence aux pseudonymes - Exemples de plan d'accès

L'utilitaire DB2 Explain permet d'afficher le plan d'accès utilisé par l'optimiseur lors du traitement des requêtes. Les exemples suivants montrent comment l'optimiseur accède aux données de pseudonymes dans un environnement qui utilise à la fois le parallélisme intrapartition et le parallélisme interpartition.

Exemple 1 : mode sécurisé

L'exemple suivant illustre une jointure entre une table locale et un pseudonyme en mode sécurisé. Le serveur fédéré traite les données locales en parallèle dans chaque partition avant de les joindre aux données de pseudonymes sur la partition coordinatrice. Le serveur fédéré ne traite pas les données de pseudonymes en parallèle sur les partitions ou les processeurs d'une partition donnée.

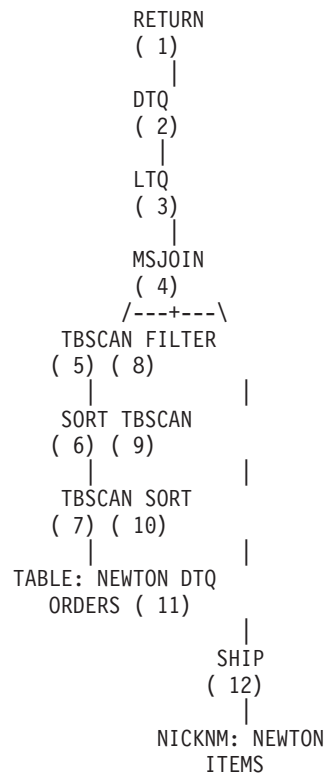
```
SELECT *  
FROM ORDERS A, ITEMS B  
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

```
          RETURN  
          ( 1)  
          |  
        HSJOIN  
          ( 2)  
    /-----+-----\  
    DTQ SHIP  
    ( 3) ( 6)  
    |           |  
    LTQ NICKNM: NEWTON  
    ( 4) ITEMS  
    |  
    TBSCAN  
    ( 5)  
    |  
    TABLE: NEWTON  
    ORDERS
```

Exemple 2 : mode isolé

L'exemple suivant illustre une jointure entre une table locale partitionnée et un pseudonyme en mode isolé. Le serveur fédéré extrait les données de pseudonymes en série sur la partition coordinatrice, puis les distribue aux autres partitions de la base de données. Ces données sont ensuite jointes en parallèle avec les données locales sur toutes les partitions de base de données appropriées. Dans chaque partition, plusieurs sous-agents lisent la table locale et joignent les données de pseudonymes. Cette opération correspond au parallélisme intrapartition, identifié dans le plan par l'opérateur LTQ. Le résultat est renvoyé à la partition coordinatrice en vue du traitement final et à l'application.

```
SELECT *  
FROM ORDERS A, ITEMS B  
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Chapitre 25. Traitement asynchrone des requêtes fédérées

L'*état asynchrone* est une méthode d'amélioration des performances des requêtes en exécutant plusieurs parties d'un plan d'accès simultanément pour réduire le temps écoulé d'une requête donnée.

Dans un système fédéré, les données sont distribuées sur les systèmes aux diverses sources de données et chaque système dispose de ses propres ressources. L'état asynchrone chevauche les opérations qui utilisent ces ressources afin que plusieurs systèmes puissent rester actifs en même temps. Le chevauchement d'opérations permet à plusieurs parties d'un plan d'accès de s'exécuter simultanément plutôt que successivement.

Les requêtes complexes qui impliquent des opérations très longues sur des sources de données éloignées peuvent bénéficier de l'état asynchrone. L'état asynchrone permet d'effectuer simultanément les types d'opérations suivants :

- Plusieurs opérations sur des sources de données éloignées
- Des opérations sur le serveur fédéré et au moins une source de données éloignée

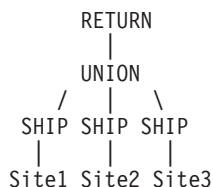
Puisque les opérations de requête consomment des ressources, l'état asynchrone peut avantager les systèmes dont les ressources sont en veille ou quand une seule des sources de données, ou le système fédéré, fonctionne à n'importe quel point de cohérence.

Traitement asynchrone des requêtes fédérées - Exemples

Les exemples de requêtes avec une jointure par fusion et une union illustrent la différence qui existe entre les opérations de requête avec et sans traitement asynchrone.

Exemple : Requête avec une opération d'union

Une simple requête effectue une opération d'union sur des données à partir de trois sources de données différentes. Le calcul nécessaire pour générer les données sur chaque source de données est très long. Le plan d'accès ressemble à ce qui suit :



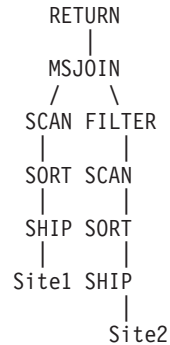
Sans état asynchrone, l'opération d'union lit les données à partir des branches de la requête, une branche à la fois, de gauche à droite. Quand les données du serveur Site1 sont lues, le serveur Site2 et le serveur Site3 sont en veille. Dans cet exemple, il faut environ 2 heures à chaque branche de l'union pour renvoyer les lignes de résultats d'un site. Le temps d'exécution total des trois branches équivaut environ à la somme de temps de chaque branche, dans cet exemple, environ six heures.

Avec l'état asynchrone, le traitement de chaque branche de l'union démarre en même temps, et les trois serveurs éloignés sont actifs simultanément. La phase d'exécution de la requête est à peu près équivalente à la phase d'exécution de la

branche la moins rapide de l'union. Dans cet exemple, la phase d'exécution est réduite à environ deux heures (presque 66 % plus rapide) par rapport aux six heures qu'il faut compter sans l'état asynchrone.

Exemple : Requête avec une opération de jointure par fusion

Une requête fusionne des données provenant de deux sources de données différentes à l'aide d'une opération de jointure par fusion (MSJOIN). Le plan d'accès de l'optimiseur ressemble à ce qui suit :



Sans état asynchrone, l'opération de jointure par fusion commence par traiter la branche la plus éloignée (gauche) et ne traite pas la branche la plus proche (droite) tant que la branche de gauche n'a pas commencé à renvoyer de lignes. Dans cet exemple, chaque branche exécute une requête complexe et, par conséquent, l'exécution est très longue. Le temps total approximatif d'exécution de la jointure par fusion équivaut à la somme de temps d'exécution de chaque branche.

Avec l'état asynchrone, les deux branches de la jointure par fusion démarrent en même temps, réduisant ainsi le temps d'exécution total de la requête.

Optimisation de l'état asynchrone

L'optimiseur de requête prend des décisions sur le traitement asynchrone des opérations éloignées dans un plan d'exécution de requête. L'*optimisation de l'état asynchrone* est le processus par lequel l'optimiseur analyse un plan d'exécution de requête existant et recherche la possibilité d'exécuter des opérations éloignées simultanément.

Plans d'accès sans état asynchrone

Dans un plan d'exécution, l'opérateur SHIP ou RPD définit une partie du plan qui est exécutée sur une source de données éloignée.

Sans état asynchrone, l'opérateur SHIP ou RPD devient actif et lance le traitement à distance uniquement quand ses données sont requises par d'autres opérateurs situés au dessus de l'opérateur SHIP ou RPD dans le plan d'exécution.

Plans d'accès optimisés pour l'état asynchrone

L'optimiseur peut permettre aux opérations éloignées définies par l'opérateur SHIP ou RPD de s'exécuter de manière asynchrone.

Dans une opération asynchrone, un opérateur de file d'attente de tables (TQ) est inséré directement sur l'opérateur SHIP ou RPD du plan d'exécution. L'opérateur TQ définit une portion du plan, appelé *sous-plan*. Le sous-plan est exécuté par une

unité d'exécution ou un processus séparé, avec sa propre mémoire. Il se lance immédiatement quand la requête démarre.

L'opérateur TQ peut être considéré comme un canal entre l'opérateur SHIP ou RPD (fournisseur de données) et l'opérateur au dessus de lui (consommateur de données) dans le plan. Ce canal dissocie l'exécution de l'opérateur SHIP du sous-plan en dessous de lui, du plan principal, et autorise l'échange de données asynchrone entre les deux sections du plan.

Un opérateur TQ, qui apparaît directement au dessus d'un opérateur SHIP ou RPD du plan, autorise les opérations éloignées définies par l'opérateur SHIP ou RPD à se lancer au début de la requête et à offrir des résultats au serveur fédéré de manière asynchrone. Quand l'état asynchrone est bénéfique à une opération éloignée donnée, l'optimiseur place un opérateur TQ directement au dessus de l'opérateur SHIP ou RPD du plan.

Les opérateurs TQ se rencontrent dans les plans d'exécution dans différentes optiques. En règle générale, un opérateur TQ désigne des opérations parallèles dans des bases de données partitionnées ou dans des bases de données activées pour le parallélisme intrapartition. Un autre type d'opérateur TQ, appelé ATQ (TQ d'état asynchrone), permet l'exécution asynchrone d'un sous-plan.

L'optimiseur rend un opérateur SHIP ou RPD donné asynchrone quand :

- les performances des requêtes doivent être améliorées
- Le nombre d'ATQ est inférieur aux limites 'par serveur' et 'par requête'
- L'opérateur n'est pas déjà asynchrone en raison de l'utilisation d'une autre technique d'optimisation
- Les restrictions relatives à l'état asynchrone ne sont pas transgressées.
- La sémantique d'une requête ne change pas

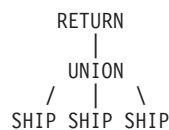
Plans d'accès - Exemples

Des exemples de plans d'accès illustrent la différence entre l'exécution du plan avec ou sans l'optimisation de l'état asynchrone.

Les deux premiers exemples montrent à quoi ressemblent les plans de jointure par fusion et d'union dans «Traitement asynchrone des requêtes fédérées - Exemples», à la page 265 quand l'état asynchrone est activé.

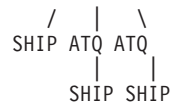
Pour plus de simplicité, les exemples montrent des plans avec des opérateurs SHIP uniquement. L'optimisation de l'état asynchrone transforme le plan de la même manière pour les opérateurs RPD que pour les opérateurs SHIP. Les opérateurs SHIP et RPD sont interchangeable, sauf indication contraire.

Exemple 1a : Plan sans état asynchrone



Exemple 1b : Plan avec état asynchrone

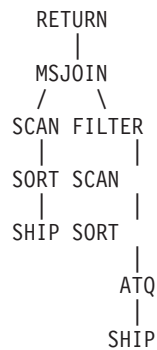




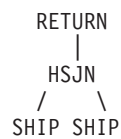
Exemple 2a : Plan sans état asynchrone



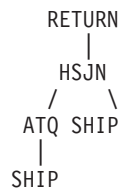
Exemple 2b : Plan avec état asynchrone



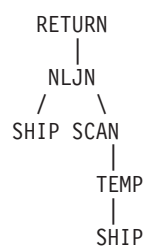
Exemple 3a : Plan sans état asynchrone



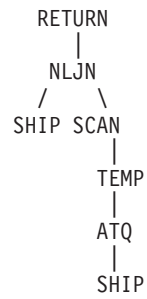
Exemple 3b : Plan avec état asynchrone



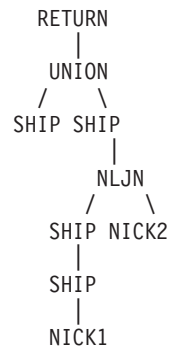
Exemple 4a : Plan sans état asynchrone



Exemple 4b : Plan avec état asynchrone

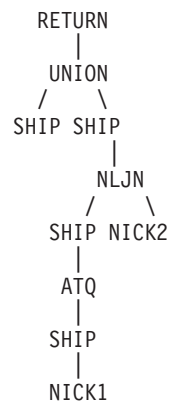


Exemple 5a : Plan sans état asynchrone

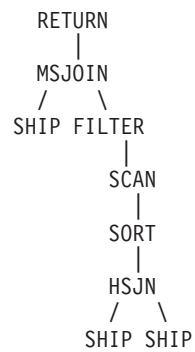


Dans ce plan, les RPD ne peuvent pas remplacer la paire SHIP-SHIP.

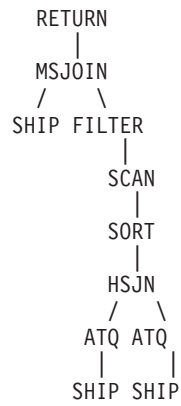
Exemple 5b : Plan avec état asynchrone



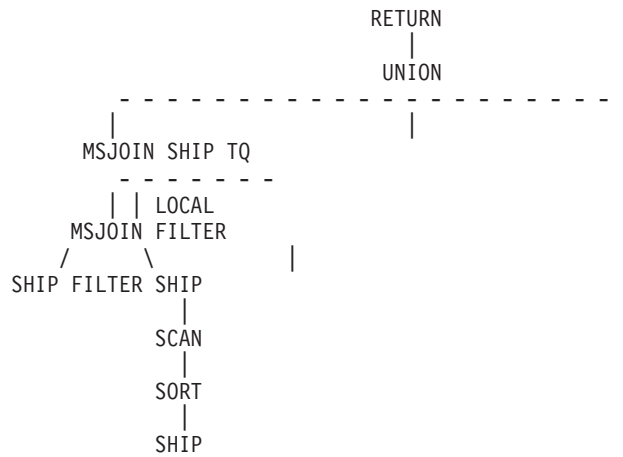
Exemple 6a : Plan sans état asynchrone



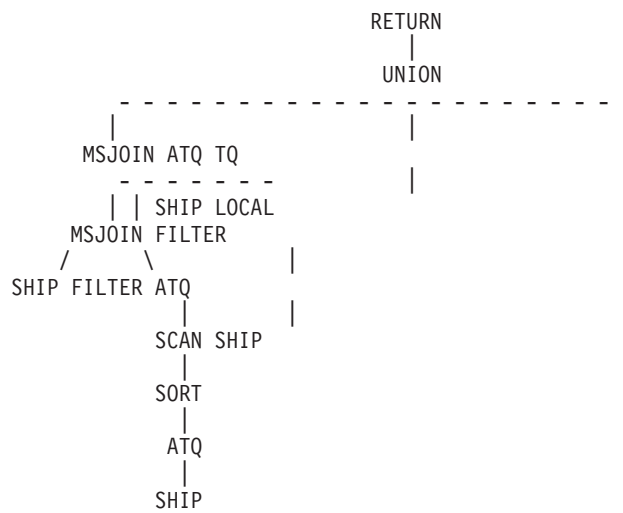
Exemple 6b : Plan avec état asynchrone



Exemple 7a : Plan sans état asynchrone



Exemple 7b : Plan avec état asynchrone



Contrôle de la consommation des ressources

Outre la possibilité d'activer l'exécution asynchrone d'une requête à distance, l'opérateur ATQ agit sur le serveur fédéré et sur les sources de données éloignées.

Puisque chaque opérateur ATQ crée un nouveau processus, et consomme de la mémoire (pour la mise en mémoire tampon), l'insertion de plusieurs ATQ dans un plan d'exécution risque d'engendrer une surconsommation des ressources système du serveur fédéré. En outre, si plusieurs opérateurs SHIP ou RPD d'une requête s'exécutent sur une source de données éloignées particulière, la mise en état asynchrone de plusieurs des opérateurs ouvre plusieurs curseurs simultanément sur cette source de données et peut produire une charge inacceptable.

Pour contrôler la consommation de ressource sur un système fédéré ou sur les sources de données, vous pouvez définir les paramètres de configuration. Les paramètres définissent le nombre total de ATQ autorisées dans une requête ainsi que le nombre total de ATQ autorisées pour chaque serveur dans une requête.

Activation de l'optimisation de l'état asynchrone

Pour activer l'optimisation de l'état asynchrone, vous devez indiquer le nombre d'opérateurs TQ d'état asynchrone pour une requête donnée et définir une option de serveur pour la source de données.

Restrictions

L'optimisation de l'état asynchrone nécessite :

- Un système fédéré avec la fonction de partitionnement de base de données DPF (Database Partitioning Feature) qui comprend plusieurs partitions de bases de données logiques.
- Un accès aux sources de données via des encapsuleurs isolés.

Cette optimisation ne prend pas en charge ces objets :

- Les pseudonymes d'une source de données dont l'accès s'effectue via un encapsuleur isolé.
- Les requêtes avec des opérations d'insertion, de mise à jour ou de suppression.

Procédure

Pour activer le traitement asynchrone, procédez comme suit :

1. Définissez un ou plusieurs des paramètres suivants :
 - Définissez le paramètre de configuration du gestionnaires de base de données FEDERATED_ASYNC sur une valeur comprise entre 0 et 32 767 incluse ou sur ANY. MAXAGENTS est un paramètre de configuration de base de données qui indique le nombre maximal de ATQ autorisées par requête. La valeur ANY permet à l'optimiseur de déterminer le nombre de ATQ pour un plan d'accès déterminé. Par défaut, la valeur est définie sur 0.
 - Facultatif : vous pouvez définir les options de liaison et de précompilation du paramètres FEDERATED_ASYNCRONY pour que les instructions statiques du module remplacent le paramètre de configuration de la requête. Par défaut, la valeur est définie sur 0.

- Facultatif : vous pouvez définir le registre spécial CURRENT FEDERATED ASYNCHRONY pour remplacer de manière dynamique la définition du paramètre de configuration du gestionnaire de base de données et l'option de définition d'accès de la requête.

Ces paramètres forment la hiérarchie suivante :

- a. Registre spécial
- b. Option de définition d'accès
- c. Paramètre de configuration du gestionnaire de bases de données

La valeur du registre spécial, si indiquée, est prioritaire sur l'option de définition d'accès, qui à son tour est prioritaire sur le paramètre de configuration du gestionnaire de bases de données.

2. Vous devez indiquer une valeur numérique pour l'option de serveur DB2_MAX_ASYNC_REQUESTS_PER_QUERY. Cette option de serveur indique le nombre maximal de requêtes asynchrones que le serveur autorise pour la requête.

L'intervalle pour l'option de serveur est compris entre -1 et 64000.

- Par défaut, la valeur est définie sur 1. Par conséquent, un opérateur SHIP ou un opérateur RPD qui appartient à un serveur peut être envisagé pour l'état asynchrone dans une requête.
- Pour la source de données ODBC uniquement, la valeur est définie par défaut sur 0. L'état asynchrone nécessite plusieurs curseurs par connexion. Cette valeur doit être définie sur 0 pour l'encapsuleur ODBC à moins que la source de données n'accepte plusieurs curseurs par connexion.

Remarques sur le réglage de l'optimisation de l'état asynchrone

Quand l'état asynchrone est activé, vous devez examiner les facteurs qui ont une incidence sur les performances.

Si votre système dispose des ressources d'unité centrale, de la mémoire et des processus suffisants, l'activation de l'état asynchrone peut améliorer les performances de vos requêtes fédérées. L'activation de l'état asynchrone peut également augmenter l'utilisation des systèmes de source éloignée par le serveur fédéré, car une source éloignée peut potentiellement traiter plus d'une requête à la fois pour une requête fédérée. Si votre système connaît des restrictions de ressources, l'état asynchrone risque de diminuer les performances.

Vous pouvez optimiser votre système pour l'état asynchrone en modifiant les paramètres de configuration afin d'obtenir le meilleur degré d'état asynchrone pour votre système.

Chaque TQ asynchrone introduit dans un plan par l'optimisation de l'état asynchrone nécessite un sous-agent supplémentaire. Si votre système dispose d'un nombre de sous-agents suffisant, vous devez envisager de régler le paramètre de configuration du gestionnaire de bases de données MAXAGENTS.

Restrictions relatives à l'optimisation de l'état asynchrone

Quand l'optimiseur applique l'optimisation de l'état asynchrone à une requête donnée, certaines restrictions s'appliquent au nombre de ATQ pouvant être utilisées dans le plan d'exécution de requête.

Le nombre d'opérateurs SHIP ou RPD qui peut être associé à une ATQ dans un plan et bénéficier de l'état asynchrone peut être supérieur au nombre maximal défini par le paramètre FEDERATED_ASYNC ou bien à la limite 'par serveur' de l'option de serveur DB2_MAX_ASYNC_REQUESTS_PER_QUERY. Dans ce cas, l'optimiseur choisit d'associer l'opérateur SHIP ou RPD à une ATQ afin que :

- Le nombre total de ATQ d'un plan soit inférieur ou égal à la valeur définie pour les paramètres suivants, dans cet ordre :
 1. registre spécial FEDERATED ASYNCHRONY, si indiqué
 2. option de définition d'accès ou de précompilation FEDERATED ASYNCHRONY, si indiquée
 3. paramètre FEDERATED_ASYNC
- Le nombre total de ATQ pour un serveur donné soit inférieur ou égal à l'option de serveur DB2_MAX_ASYNC_REQUESTS_PER_QUERY pour ce serveur.
- Les opérateurs SHIP ou RPD concernés peuvent profiter de cette association à une ATQ et améliorer les performances des requêtes.

Indication de l'application ou non de l'optimisation de l'état asynchrone à une requête

Pour déterminer si l'optimisation de l'état asynchrone est appliquée à une requête donnée, plusieurs méthodes permettent de vérifier le plan d'accès de la requête et un opérateur spécifique.

Vous pouvez vérifier toutes les sorties suivantes de la requête en question :

- sortie db2exfmt
- sortie Visual Explain
- sortie dynexpln

Chaque sortie affiche les demandes de requêtes asynchrones dans le plan d'accès comme des ATQ. La propriété 'origine' de la description de l'ATQ indique 'Etat asynchrone'.

Le fragment de plan suivant présente l'utilisation de l'opérateur ATQ et affiche ses propriétés détaillées.

```

                                1.6e+06
                                HSJOIN
                                ( 2)
                                4213.74
                                131
                                /-----+-----\
                                40000 1000
                                HSJOIN SHIP
                                ( 3) ( 10)
                                1122.26 427.733
                                117 14
                                /-----+-----\
                                1000 1000 1000
                                ATQ          ATQ          NICKNM: NEWTON
                                ( 4) ( 7) S1_NN07
                                511.795 532.669
```

```

      16 101
      |           |
    1000 1000
    SHIP SHIP
    ( 5) ( 8)
    478.773 499.887
      16 101
      |           |
    1000 1000
NICKNM: NEWTON NICKNM: NEWTON
S2_NN02 S3_NN15

```

```

- montre l'origine de l'ATQ comme ASYNCHRONY :
4) TQ : (Table Queue)
Coût total cumulé : 511,795
Coût d'UC cumulé : 2,79486e+06
Coût d'E-S cumulé : 16
Coût total recalculé cumulé : 68,9489
Coût d'UC recalculé cumulé : 1,72372e+06
Coût d'E-S recalculé cumulé : 0
Coût première ligne du tableau cumulé : 30,8308
Coût de com. cumulé : 18,2176
Coût de première com. cumulé : 0
Estimation des mémoires tampons en pool : 16
Coût de communication à distance : 538.297

```

Arguments :

```

-----
JN INPUT: (branche d'entrée de jointure)
  OUTER
LISTENER: (Type de file d'attente de tables de l'écouteur)
  FALSE
TQMERGE : (Indicateur de fusion de la file d'attente de tables)
  FALSE
TQORIGIN: (Type d'origine de la file d'attente de tables)
ASYNCHRONY
TQREAD : (Type de lecture de la file d'attente de tables)
  READ AHEAD
TQSEND : (Type d'écriture de la file d'attente de tables)
  DIRECTED
UNIQUE : (Indicateur obligatoire d'unicité)
  FALSE

```

Si vous ne trouvez pas d'opérateurs ATQ dans le plan d'accès d'une requête, vérifiez que les conditions suivantes sont remplies :

- Le système est un système fédéré activé pour la fonction DPF.
- La requête accède aux pseudonymes sur une source de données dont l'accès s'effectue via un encapsuleur isolé.
- L'état asynchrone est activé à l'aide du paramètre de configuration du gestionnaire de bases de données `FEDERATED_ASYNC`, du registre spécial `CURRENT FEDERATED ASYNCHRONY` ou de l'option de définition d'accès `FEDERATED_ASYNC`.
- L'option de serveur `DB2_MAX_ASYNC_REQUESTS_PER_CONNECTION` est définie sur une valeur différente de zéro pour les pseudonymes au niveau de chaque serveur.

Chapitre 26. Optimisation globale

Le compilateur SQL fonctionne en trois phases qui permettent d'obtenir une stratégie d'accès optimale pour l'évaluation d'une requête qui référence une source de données éloignée. Ces phases sont les suivantes : l'analyse du transfert de l'exécution sur la base de données, l'optimisation globale et la génération SQL à distance.

Pour une requête soumise à la base de données fédérée, la stratégie d'accès doit impliquer le fractionnement de la requête originale en un ensemble de fragments, puis ensuite l'association des résultats de ces fragments de requête.

En utilisant le résultat de la phase d'analyse du transfert de l'exécution sur une base de données comme une recommandation, l'optimiseur de requête décide de l'endroit de l'évaluation de chaque opération. Une opération peut être évaluée localement sur le serveur fédéré ou à distance sur la source de données. La décision dépend du résultat du niveau du modèle de coût utilisé par l'optimiseur. Ce modèle détermine :

- Le coût de l'évaluation de l'opération
- Le coût de transmission des données ou des messages entre le serveur fédéré et les sources de données

L'objectif de l'optimisation globale est de produire un plan d'accès qui optimise les opérations de requête sur toutes les sources de données globalement, parmi les systèmes fédérés. Un plan d'accès optimal sous tous ses aspects est le plus économique en termes d'exécution sur un système fédéré. Le revers de la phase de génération SQL à distance est la conversion de la totalité du plan optimal en fragments de requête qui sont exécutés par des sources de données individuelles.

Le compilateur SQL dispose d'une base de connaissances qui contient les caractéristiques des sources de données prises en charge et les métadonnées relatives aux données de ces sources de données. L'optimiseur ne génère pas de SQL, de fragments de requête ou d'indications sur les plans que la source de données éloignée ne peut comprendre ou accepter.

De nombreux facteurs sont déterminants dans le résultat de l'optimisation globale et par conséquent, peuvent avoir une incidence sur les performances de la requête. Les principaux facteurs sont les caractéristiques de serveur et les caractéristiques de pseudonymes.

Les encapsuleurs relationnels et non relationnels disposent d'une méthode d'élaboration de plan d'accès différente, mais le concept et l'effet final sont identiques.

Caractéristiques de serveur ayant une incidence sur l'optimisation globale

Lors de la création ou de la modification de la définition d'un serveur, certaines options sont déterminantes dans les performances de la requête.

Vous fournissez à l'optimiseur de requêtes des informations relatives aux caractéristiques de serveur de la source de données via les paramètres de l'option

de serveur. Les paramètres de l'option de serveur font partie de la définition du serveur de la source de données. Vous pouvez définir les options de serveur dans une instruction CREATE SERVER, quand vous établissez au départ la définition de serveur. L'instruction ALTER SERVER permet d'ajouter des options de serveur à une définition de serveur existante. Les paramètres d'option de serveur sont stockés dans le catalogue global de la base de données fédérée.

Ces options sont classées en options de localisation (le nom de l'ordinateur de la source de données), de sécurité (les informations d'authentification) et de performances (le rapport de l'unité centrale).

Les options de performance permettent à l'optimiseur de déterminer si l'évaluation d'une opération peut s'effectuer au niveau de la source de données et si cette évaluation permet d'accélérer l'exécution. Les options de serveur ayant une incidence sur les performances qui risquent de nécessiter un réglage sont les suivantes :

- CPU_RATIO
- IO_RATIO
- COMM_RATE
- COLLATING_SEQUENCE
- PLAN_HINTS
- VARCHAR_NO_TRAILING_BLANKS
- NO_EMPTY_STRING

Le réglage des options de serveur CPU_RATIO, IO_RATIO ou COMM_RATE doit être effectué avec prudence car vous risquez d'obtenir des erreurs imprévues si le calcul du coût d'une requête entraîne une utilisation partielle ou un dépassement de capacité. La plupart du temps, les valeurs par défaut de ces options sont suffisantes. En règle générale, la vérification de l'exactitude des statistiques relatives aux objets référencés dans vos requêtes est plus importante que le réglage des valeurs de ces options de serveur.

Rapport relatif de la vitesse de l'unité centrale

Cette valeur indique le rapport entre la vitesse de l'unité centrale du serveur fédéré et celle de l'unité centrale du serveur sur lequel se trouve la source de données éloignée.

Cette valeur est égale à la vitesse de l'unité centrale du serveur fédéré divisée par la vitesse de l'unité centrale du serveur de la source de données éloignée. Par exemple, si la vitesse de l'UC du serveur fédéré est deux fois plus rapide que celle de l'UC du serveur éloigné, alors CPU_RATIO doit indiquer 2. Si la vitesse de l'UC du serveur fédéré équivaut à seulement un tiers de la vitesse de l'UC du serveur éloigné, alors CPU_RATIO doit indiquer 0,33.

Si l'option de serveur relative au rapport de l'UC n'est pas explicitement définie, l'optimiseur fédéré utilise par défaut la valeur 1, qui indique que la vitesse de l'UC fédérée et la vitesse de l'UC de la source de données sont égales.

Un rapport bas indique que l'UC du serveur source est plus rapide que l'UC du serveur fédéré. Dans les cas de rapports bas, l'optimiseur envisagera d'effectuer les opérations de transfert de l'exécution qui sont exigeantes en termes d'UC sur la source de données. Un rapport bas est un rapport dont la valeur est inférieure à 1.

Rapport relatif de la vitesse d'entrée-sortie

Cette valeur indique le rapport entre la vitesse d'entrée-sortie du serveur fédéré et la vitesse d'entrée-sortie du serveur sur lequel se trouve la source de données éloignée.

Cette valeur est égale au débit d'entrée-sortie du serveur fédéré divisé par le débit d'entrée-sortie du serveur éloigné. Par exemple, si le débit d'entrée-sortie du serveur fédéré est deux fois plus important que le débit d'entrée-sortie du serveur éloigné, IO_RATIO doit indiquer 2. Mais si le débit d'entrée-sortie du serveur fédéré est deux fois moins important que celui du serveur éloigné, alors IO_RATIO doit indiquer 0,5.

Si l'option de serveur relative au rapport d'entrée-sortie n'est pas explicitement définie, l'optimiseur fédéré utilise par défaut la valeur 1, qui indique que les débits d'entrée-sortie des serveurs fédéré et éloigné sont égaux.

Un rapport d'IO_RATIO bas, inférieur à un, indique que le débit d'entrée-sortie du serveur éloigné est supérieur à celui du serveur fédéré. Dans ce cas, l'optimiseur va plutôt favoriser la transfert de l'exécution d'opérations exigeantes en termes d'entrée-sortie vers la source de données éloignée. Un rapport bas est un rapport dont la valeur est inférieure à 1.

Débit de communication entre le serveur fédéré et la source de données

Un débit de communication bas indique que les communications réseau entre le serveur fédéré et la source de données sont lentes.

Le paramètre de l'option de serveur COMM_RATE détermine le débit de communication. Il représente la vitesse de la connexion réseau entre le serveur de la source de données et le serveur fédéré. Le débit se mesure en mégabits par seconde. Par défaut, la valeur est définie sur 2 Mbit/s.

Les débits de communication bas encouragent l'optimiseur de requête à réduire le nombre de messages envoyés vers ou provenant de cette source de données. Si l'option de serveur COMM_RATE est définie sur un très petit nombre, l'optimiseur produit une requête qui nécessite un trafic réseau minimal.

Séquences de classement de source de données

La séquence de classement que vous avez sélectionnée risque d'avoir une incidence sur les performances de la base de données fédérée. L'option de serveur COLLATING_SEQUENCE permet d'indiquer si la séquence de classement d'une source de données correspond à la séquence de classement de la base de données fédérée locale.

Le serveur fédéré peut transférer l'exécution du traitement dépendant de l'ordre qui implique des données de type caractères sur la source de données, si l'option de serveur COLLATING_SEQUENCE indique que la séquence de classement de la source de données et celle de la base de données fédérée correspondent. Si la séquence de classement de la source de données ne correspond pas à la séquence de classement de la base de données fédérée, l'optimiseur considère les données qui sont extraites de cette source de données comme étant non ordonnées. La base de données fédérée va extraire les données pertinentes et traiter localement toutes les données de type caractères qui dépendent de l'ordre, ce qui risque de ralentir la requête et de réduire les performances.

Indications de plan éloigné

Les indications de plan sont des fragments d'instructions qui fournissent des informations complémentaires aux optimiseurs de source de données.

L'option de serveur `PLAN_HINTS` permet de générer des indications de plan éloigné. Pour certains types de requête, ces informations permettent d'améliorer les performances des requêtes. Les indications de plan peuvent aider l'optimiseur de source de données à décider s'il doit utiliser un index, ou le type d'index ou de séquences de jointure de table à utiliser.

Vous devez exécuter certains tests pour déterminer si cette option de serveur va améliorer les performances de vos requêtes.

Vous ne pouvez pas coder vos propres indications de plan dans une requête.

Si les indications de plan sont activées, la requête envoyée à la source de données contient des informations complémentaires. Par exemple, une instruction envoyée à un optimiseur Oracle avec des indications de plan pourrait ressembler à ce qui suit :

```
SELECT /*+ INDEX (table1, t1index)*/  
  col1  
FROM table1
```

L'indication de plan correspond à la chaîne `/*+ INDEX (table1, t1index)*/`

Caractéristiques des pseudonymes ayant une incidence sur l'optimisation globale

Plusieurs facteurs propres aux pseudonymes sont déterminants dans l'optimisation globale, y compris les informations d'indexation et les statistiques du catalogue global.

Il est important que les informations d'indexation et les données statistiques du catalogue global disponibles dans le compilateur SQL soient à jour.

Spécifications de l'index

Le compilateur SQL s'appuie sur les informations d'indexation pour optimiser les requêtes.

Les informations d'indexation d'une table de source de données existent uniquement quand un pseudonyme est créé pour cette table. Une fois le pseudonyme créé, toute modification apportée à l'index sur cette table de source de données n'est pas mise à jour sur le serveur fédéré. Quand les informations d'indexation éloignées changent, vous pouvez mettre à jour les informations d'indexation stockées sur le serveur fédéré en supprimant le pseudonyme de la table puis en le créant de nouveau. Sinon, si un nouvel index est ajouté à la table de source de données, vous pouvez définir une spécification d'index pour le pseudonyme sur le serveur fédéré.

Les informations d'indexation ne sont pas rassemblées pour les pseudonymes des objets qui ne disposent pas d'index, tels que les objets de source de données non relationnelle, les synonymes ou les vues.

Si un objet avec un pseudonyme défini n'a pas d'index, vous pouvez lui créer une spécification d'index. Les spécifications d'index élaborent une définition d'index

dans le catalogue global. La spécification d'index ne correspond pas à un index réel. L'instruction CREATE INDEX, avec la clause SPECIFICATION ONLY, permet de créer une spécification d'index. La syntaxe de création d'une spécification d'index sur un pseudonyme est semblable à la syntaxe de création d'un index sur une table locale.

Vous devez envisager de créer des spécifications d'index dans les situations suivantes :

- Un nouvel index est ajouté à une table.
- Vous créez un pseudonyme pour un objet de source de données qui ne contient pas d'index tel qu'une vue ou un synonyme.

Quand vous créez une spécification d'index (SPECIFICATION ONLY) sur un pseudonyme et indiquez que l'index est unique, la base de données fédérée ne vérifie pas que les valeurs de colonne de la table éloignée sont uniques. Si elles ne le sont pas, les requêtes effectuées sur les pseudonymes qui contiennent cette colonne d'index risquent de renvoyer des données incorrectes ou des résultats erronés.

Déterminez bien vos besoins avant de lancer les instructions CREATE INDEX...SPECIFICATION ONLY sur un pseudonyme pour une vue de source de données :

- Si la vue éloignée est une simple instruction SELECT sur une table de source de données avec un index, la création d'une spécification d'index sur le pseudonyme correspondant à l'index de la table de source de données peut améliorer considérablement les performances de la requête.
- Si une spécification d'index est créée pour une vue éloignée qui n'est pas une simple instruction SELECT (une vue créée par la fusion de deux tables, par exemple), les performances de la requête peuvent être réduites.

Par exemple, considérons une spécification d'index créée pour une vue éloignée, correspondant à la fusion de deux tables. L'optimiseur peut choisir cette vue comme l'élément interne d'une jointure en boucle imbriquée. Les performances de la requête risquent d'être réduites car la jointure sera évaluée plusieurs fois. Sinon, l'autre possibilité est de créer des pseudonymes pour chacune des tables référencées dans la vue de source de données et de créer une vue fédérée qui référence les deux pseudonymes.

Statistiques du catalogue global

Le catalogue global d'un serveur fédéré contient des informations statistiques qui permettent d'optimiser les requêtes.

Le serveur fédéré s'appuie sur les statistiques des objets de source de données, dont les pseudonymes ont été définis, pour optimiser les requêtes qui impliquent ces pseudonymes. Ces statistiques sont extraites de la source de données quand vous créez un pseudonyme pour un objet de source de données à l'aide de l'instruction CREATE NICKNAME. La base de données fédérée vérifie la présence de l'objet sur la source de données, puis tente de regrouper les données statistiques existantes de la source de données. Les informations utiles à l'optimiseur sont lues dans les catalogues de la source de données puis insérées dans le catalogue global du serveur fédéré. Tout ou partie des informations du catalogue de la source de données pouvant être utilisé par l'optimiseur, nous vous conseillons de mettre à jour les statistiques (à l'aide de la commande de source de données équivalente à RUNSTATS) sur la source de données avant de créer un pseudonyme.

Les statistiques de catalogue décrivent la taille globale des tables et vues, ainsi que l'intervalle de valeurs dans les colonnes associées. Les informations extraites incluent, entre autres :

- Le nombre de lignes dans un objet de pseudonyme
- Le nombre de pages nécessaires à un pseudonyme
- Le nombre de valeurs distinctes dans chaque colonne d'une table
- Le nombre de valeurs distinctes dans les colonnes d'un index
- Les valeurs maxi/mini d'une colonne

La base de données fédérée peut extraire les données statistiques conservées dans une source de données, mais ne détecte pas automatiquement si les données statistiques existantes des sources de données ont été mises à jour. En outre, la base de données fédérée n'a pas la possibilité de traiter les modifications apportées à la définition ou à la structure des objets au niveau des sources de données (lors de l'ajout d'une colonne à une table, par exemple).

Si les caractéristiques de structure ou de données statistiques d'un objet distant qui contient un pseudonyme sont modifiées, vous avez plusieurs possibilités de mettre à jour les statistiques :

- Collecte manuelle de statistiques
 - Exécutez la commande équivalente à RUNSTATS sur la source de données. Puis supprimez le pseudonyme actuel et créez-en un de nouveau. Nous vous recommandons cette méthode pour la mise à jour des statistiques.
L'avantage de cette méthode, outre la mise à jour des statistiques, est que toutes les informations relatives aux nouvelles modifications de structure et d'index apportées à l'objet éloigné vont se refléter sur le nouveau pseudonyme. Par contre, elle a l'inconvénient d'annuler toutes les vues ou tous les modules basés sur l'ancien pseudonyme.
 - Utilisez la fonction de mise à jour des statistiques des pseudonymes du Centre de contrôle DB2. Sinon, vous pouvez utiliser la procédure mémorisée sous-jacente SYSPROC.NNSTAT(), disponible à partir du processeur de la ligne de commande.
La fonction de mise à jour des statistiques des pseudonymes (ou SYSPROC.NNSTAT()) actualise uniquement les statistiques des pseudonymes ; elle ne modifie pas le pseudonyme pour l'adapter à une quelconque modification apportée à la structure d'un objet éloigné. Par exemple, si l'objet éloigné compte une nouvelle colonne, la mise à jour des statistiques n'ajoute de colonne au pseudonyme.
 - Vous pouvez mettre à jour manuellement les statistiques dans la vue catalogue SYSSTAT.TABLES. Utilisez cette méthode uniquement si les informations statistiques de la source de données éloignée sont incorrectes et incomplètes.
- Collecte automatique de statistiques
Cette fonction s'exécute par défaut pour améliorer les performances en collectant de manière automatisée les statistiques de pseudonymes et de tables à jour.

Mise à jour des modifications apportées aux lignes

Si un grand nombre de lignes est ajouté ou supprimé d'un objet de la source de données, la base de données fédérée n'a pas connaissance de ces modifications car les statistiques de catalogue du pseudonyme continuent d'indiquer l'ancien nombre de lignes.

Toutefois, vous devriez noter des pertes de performance car l'optimiseur continue de prendre des décisions en s'appuyant sur des informations de statistiques de pseudonymes qui ne sont plus exactes. Après avoir mis à jour les statistiques de l'objet éloigné au niveau de la source de données, vous pouvez actualiser les statistiques du pseudonyme pour que l'optimiseur utilise des statistiques correctes quand il génère et sélectionne les plans d'accès pour le traitement de requêtes sur la source de données.

Mise à jour des statistiques lors de modifications apportées aux colonnes

Quand des modifications sont apportées à la structure d'un objet de source de données (quand une colonne est ajoutée à une table, par exemple), vous devez suivre plusieurs étapes pour mettre à jour les statistiques de cet objet dans le catalogue de la base de données fédérée.

A propos de cette tâche

Si des colonnes sont ajoutées, supprimées ou modifiées sur la source de données, vous risquez d'obtenir des résultats incorrects ou de recevoir un message d'erreur. Par exemple, supposons que le pseudonyme *EUROSALES* renvoie à la table *europe* d'une base de données Sybase. Si une nouvelle colonne appelée *CZECH* est ajoutée à la table, la base de données fédérée ne reconnaîtra pas l'existence de cette colonne *CZECH*. Les requêtes qui font référence à cette colonne vont renvoyer un message d'erreur.

Procédure

Pour mettre à jour les statistiques pour cet objet quand une modification est apportée à une colonne, procédez comme suit :

1. Lancez la fonction sur la source de données qui équivaut à la commande DB2 RUNSTATS. Cette action va actualiser les statistiques stockées dans le catalogue de source de données.
2. Supprimez le pseudonyme en cours de l'objet de source de données à l'aide de l'instruction DROP NICKNAME.
3. Créez de nouveau le pseudonyme à l'aide de l'instruction CREATE NICKNAME.

Analyse de l'optimisation globale

Les informations détaillées relatives aux plans d'accès, y compris certaines informations utilisées par l'optimiseur global pour sélectionner le plan optimal, sont conservées dans des tables Explain, séparées du plan d'accès actuel même.

Ces informations permettent d'effectuer une analyse approfondie d'un plan d'accès. Les tables Explain, accessibles sur tous les systèmes d'exploitation pris en charge, contiennent des informations relatives aux instructions SQL dynamiques et statiques. Vous pouvez accéder aux tables Explain à l'aide des instructions SQL. Cette action simplifie ainsi la manipulation du résultat, les comparaisons entre les différentes requêtes ou les comparaisons d'une même requête sur une période prolongée.

Plusieurs méthodes permettent d'extraire les informations du plan d'accès global des tables Explain :

- Vous pouvez utiliser l'outil de format table Explain, db2exfmt, pour présenter les informations extraites des tables Explain dans un format prédéfini.
- Vous pouvez également utiliser la fonction Explain intégrée dans le Centre de contrôle DB2 conjointement avec Visual Explain pour comprendre le plan d'accès qui a été sélectionné pour une instruction SQL particulière.

Cette fonction Explain permet d'explicitier à la fois les instructions SQL dynamiques et statiques. La différence entre Visual Explain et db2exfmt se trouve dans la présentation des informations, puisque Visual Explain le fait dans un format graphique et db2exfmt au format texte. Les deux méthodes fournissent le même niveau de détail.

- Les outils db2expln et dynexpln permettent de comprendre le plan d'accès qui est sélectionné pour une instruction SQL particulière.

Pour une compréhension totale des résultats de db2exfmt, Visual Explain, db2expln ou dynexpln, vous devez connaître :

- Les différentes instructions SQL prises en charge et la terminologie liées à ces instructions (telles que les prédicats dans une instruction SELECT)
- Le but d'un module (plan d'accès)
- L'objectif et le contenu des tables du catalogue système
- Les opérateurs de traitement de requête de base tels que les jointures, le groupage, l'agrégation et les tris

Présentation des décisions d'optimisation de plan d'accès

Cette section établit la liste des questions classiques liées à l'optimisation ainsi que les points à étudier pour améliorer les performances.

Pourquoi une jointure entre deux pseudonymes d'une même source de données n'est-elle pas évaluée à distance ?

Vous pouvez vérifier les éléments d'une opération de jointure, les prédicats de jointures et le nombre de lignes du résultat afin de connaître les raisons pour lesquelles une jointure entre deux pseudonymes de la même source de données n'est pas évaluée à distance

Les points à étudier sont les suivants :

- Opérations de jointure. La source de données prend-elle en charge les jointures ?
- Prédicats de jointure. Le prédicat de jointure peut-il être évalué à la source de données éloignée ?
- Nombre de lignes du résultat de jointure. Vous pouvez déterminer le nombre de lignes à l'aide de Visual Explain. La jointure produit-elle un ensemble de lignes plus important que deux pseudonymes associés ? Les nombres correspondent-ils à la réalité ? Si la réponse est non, vous devez envisager de mettre à jour les statistiques de pseudonymes à l'aide de la procédure mémorisée SYSPROC.NNSTAT().

Pourquoi l'opérateur GROUP BY n'est-il pas évalué à distance ?

Les points à étudier sont les suivants :

- Syntaxe de l'opérateur. Vérifiez que l'opérateur peut être évalué sur la source de données éloignée.

- Nombre de lignes. Vérifiez le nombre estimé de lignes dans les entrées et sorties de l'opérateur GROUP BY à l'aide de Visual Explain. Ces deux nombres sont-ils très proches ? Si la réponse est oui, l'optimiseur considère qu'il est plus économique d'évaluer cet opérateur GROUP BY localement. De même, ces deux nombres reflètent-ils la réalité ? Si la réponse est non, vous devez envisager de mettre à jour les statistiques de pseudonymes à l'aide de la procédure mémorisée SYSPROC.NNSTAT().

Pourquoi l'instruction n'est-elle pas intégralement évaluée à distance ?

Le serveur fédéré cherche à garantir une parfaite analogie entre la sémantique de la requête et les résultats obtenus pour les requêtes fédérées comme si elles avaient été intégralement évaluées par DB2 Database for Linux, UNIX et Windows. La phase d'analyse du transfert de l'exécution du compilateur de requête permet de déterminer si le transfert du traitement vers des sources éloignées conservera la sémantique DB2. Ainsi, l'exécution des opérations de requête fédérée peut être transférée en toute sécurité uniquement si les opérations correspondantes sur la source éloignée ont la même signification et les mêmes résultats. Le plus souvent, le transfert total du traitement d'une requête sur une seule source éloignée est impossible à cause de l'existence de différences subtiles dans les fonctionnalités entre le serveur fédéré et la source éloignée pour une ou plusieurs opérations dans la requête.

L'optimiseur effectue une optimisation basée sur le coût. Même si l'analyse de transfert de l'exécution sur la base de données indique que chaque opérateur peut être évalué au niveau de la source de données éloignée, l'optimiseur s'appuie toujours sur ses estimations de coût pour générer un plan optimal global. De nombreux facteurs contribuent à la prise de décision sur le choix de ce plan. Supposons que la source de données éloignée traite toutes les opérations dans la requête d'origine. Toutefois, sa vitesse d'unité centrale est beaucoup plus lente que celle du serveur fédéré. Il peut s'avérer plus avantageux d'effectuer les opérations au niveau du serveur fédéré. Si vous n'obtenez pas les performances attendues, vérifiez les statistiques de serveur dans la table du catalogue SYSSTAT.SERVEROPTIONS.

Pourquoi un plan généré par l'optimiseur et intégralement évalué à distance obtient-il de moins bonnes performances que la requête originale exécutée directement sur la source de données éloignée ?

Les points à étudier sont les suivants :

- L'instruction SQL éloignée générée par l'optimiseur de requête. Outre le remplacement des pseudonymes par des noms de tables éloignées correspondant, l'instruction SQL éloignée générée diffère en règle générale de l'instruction fédérée originale des façons suivantes :
 - Le classement des prédicats de la requête peut avoir changé.
 - Les prédicats trouvés dans la requête originale peuvent avoir été supprimés, remplacés par des prédicats équivalents, ou augmentés par des prédicats supplémentaires.
 - Des sous-requêtes peuvent avoir été réécrites comme des jointures.
 - Des fonctions supplémentaires qui effectuent la conversion ou la troncature de chaîne peuvent avoir été ajoutées pour conserver la sémantique DB2

A l'exception du dernier élément mentionné ci-dessus, ces modifications ont souvent une incidence favorable sur les performances. Toutefois, dans certains cas, les modifications peuvent entraîner l'optimiseur de requête éloignée à générer un plan différent (et plus lent) comme il l'aurait fait pour une requête originale

Un bon optimiseur de requête ne devrait pas dépendre du classement des prédicats d'une requête. Malheureusement, les optimiseurs DBMS ne sont pas tous identiques. Il est probable que l'optimiseur de la source de données éloignée génère un plan différent, basé sur le classement du prédicat d'entrée. Si tel est le cas, il s'agit d'un problème inhérent à l'optimiseur éloigné. Vous devez envisager de modifier le classement des prédicats ou de contacter le service de maintenance de la source de données éloignée pour demander une assistance.

De même, vérifiez les prédicats de remplacement. Un bon optimiseur de requête ne devrait pas dépendre des prédicats de remplacement équivalents. Il est possible que l'optimiseur de la source de données éloignée génère un plan différent, basé sur le prédicat d'entrée. Par exemple, certains optimiseurs ne peuvent pas générer d'instructions de jointure transitive pour les prédicats.

- Le nombre de lignes renvoyées. Vous pouvez obtenir ce nombre à l'aide de Visual Explain. Si la requête renvoie un grand nombre de lignes, le trafic réseau est un goulot d'étranglement potentiel.
- Fonctions supplémentaires. L'instruction SQL éloignée contient-elle davantage de fonctions que la requête originale ? Certaines des fonctions supplémentaires peuvent être générées pour convertir des types de données. Vérifiez qu'elles sont nécessaires.

Chapitre 27. Éléments du contrôleur système qui ont une incidence sur les performances

Le contrôleur système de la base de données fédérée collecte les statistiques relatives à l'état en cours du gestionnaire de base de données et les données d'activité telles que les compteurs et autres mesures du traitement de la base de données.

Dans un système fédéré, le contrôleur système de la base de données permet de collecter des données sur l'activité de la base de données, ainsi que sur les performances du système et des applications.

Le commutateur du moniteur d'horodatage permet d'effectuer le suivi des temps de réponse des interactions entre la base de données fédérée et une source de données. Les éléments de données fédérées suivis par le commutateur du moniteur d'horodatage sont les suivants :

- Temps de réponse des créations de pseudonymes
- Temps de réponse des suppressions
- Temps de réponse des insertions
- Durée des fonctions passe-système
- Temps de réponse des requêtes
- Durée des verrouillages à distance
- Durée des procédures mémorisées
- Temps de réponse de mise à jour

La valeur par défaut du commutateur du moniteur d'horodatage est ON.

Recommandation : Vous pouvez améliorer les performances en commutant le moniteur d'horodatage pour toutes les applications sur OFF. Si pour une application, la valeur de ce commutateur est ON, le système continue de collecter les temps de réponse. Par conséquent, les performances ne sont pas améliorées si vous ne désactivez le commutateur du moniteur d'horodatage que pour certaines de vos applications.

La désactivation de ce commutateur n'a pas d'autres implications.

- Pour désactiver le commutateur du moniteur d'horodatage pour l'ensemble des applications, vous devez arrêter, puis redémarrer l'instance DB2 afin d'implémenter la modification.
- La désactivation du commutateur du moniteur d'horodatage désactive la collecte des données d'horodatage pour les applications fédérées ou non. La base de données locale ne reçoit pas non plus de données d'horodatage.

Si vous avez besoin de données d'horodatage pour des applications non fédérées locales, vous ne devez pas désactiver ce commutateur.

Vous pouvez désactiver le commutateur du moniteur d'horodatage pour toutes les applications à l'aide de la commande suivante :

```
update dbm cfg using dft_mon_timestamp off
```

Exécutez ensuite :

```
db2stop  
db2start
```

L'arrêt puis le démarrage du serveur fédéré garantit que le commutateur est désactivé pour l'ensemble des applications.

Les informations spécifiques à chacun des éléments suivis par le commutateur du moniteur d'horodatage font l'objet d'une autre rubrique.

Chapitre 28. Tables de requêtes matérialisées

Une table de requêtes matérialisée est une table qui met en mémoire cache les résultats d'une requête. Quand vous soumettez de nouveau la requête, le moteur de base de données peut renvoyer les données depuis la table de requête matérialisée. Les tables de requêtes matérialisées avec pseudonymes permettent d'améliorer les performances de vos requêtes.

Les tables de requêtes matérialisées sont utilisées lors de la création d'une table de mémoire cache. La table de mémoire cache stocke les données locales définies par les tables de requêtes matérialisées qui lui sont associées.

Tables de requêtes matérialisées et systèmes fédérés - Généralités

Une table de requêtes matérialisée est une table qui met en mémoire cache les résultats d'une requête. Quand vous soumettez de nouveau la requête, le moteur de base de données peut renvoyer les données depuis la table de requête matérialisée au lieu de répéter le traitement de la requête.

Vous pouvez vous servir des tables de requêtes matérialisées avec les pseudonymes pour améliorer les performances d'une requête et pour encapsuler un composant de logique. Les tables de requêtes matérialisées sont utilisées lors de la création de tables de mémoire cache.

L'optimiseur SQL détermine si une requête sera exécutée de manière plus efficace avec une table de requête matérialisée qu'avec des pseudonymes ou des tables de base. L'optimiseur utilise les facteurs suivants pour sélectionner une table de requête matérialisée :

- La table de requêtes matérialisée doit correspondre à tout ou partie de la requête.
- Le critère de la période de régénération doit être rempli.
- Le plan d'accès qui utilise une table de requête matérialisée doit être plus économique que celui qui utilise les pseudonymes ou les tables de base.

Les tables de requêtes matérialisées qui impliquent des pseudonymes pour les objets des sources de données suivants sont prises en charge :

- Sources de données relationnelles
 - DRDA
 - Informix
 - JDBC
 - ODBC
 - Oracle
 - Sybase
 - Microsoft SQL Server
 - Teradata
- Sources de données non relationnelles
 - BioRS
 - Excel
 - Fichiers structurés sous forme de table
 - Web Services
 - XML

Création d'une table de requêtes matérialisées fédérées

Les tables de requêtes matérialisées permettent de placer les données en mémoire cache locale et d'améliorer les performances de vos requêtes. Vous pouvez utiliser des pseudonymes de sources de données relationnelles et non relationnelles pour créer des tables de requêtes matérialisées.

Restrictions

- «Restrictions propres aux sources de données des tables de requêtes matérialisées»
- Si une requête contient un modèle de fonction dans un prédicat ou une liste de sélection, ce modèle de fonction doit faire partie de la table de requêtes matérialisée.
- «Restrictions sur l'utilisation des tables de requêtes matérialisées avec les pseudonymes», à la page 289

Procédure

Pour créer une table de requêtes matérialisée, exécutez une instruction CREATE TABLE qui fait référence aux pseudonymes représentant les objets de source de données à inclure.

Vous pouvez alimenter une table de requêtes matérialisée gérée par les utilisateurs en utilisant une instruction INSERT dans une instruction de sous-requête. Par exemple :

```
insert into my_mqt (select ..from n1, n2 where ..)
```

la portion de sélection de la requête correspondant à la définition de la table de requêtes matérialisée. L'optimiseur peut utiliser my_mqt pour remplacer la portion de sélection de la requête. Dans ce cas, l'instruction devient :

```
insert into my_mqt (select .. from my_mqt);
```

La table de requêtes matérialisée devient alors la source de l'opération d'insertion. Pour empêcher cela, vous pouvez exécuter l'une des commandes suivantes pour désactiver temporairement la table de requêtes matérialisée :

```
SET CURRENT REFRESH AGE 0  
SET CURRENT MAINTAINED TABLE TYPE FOR OPTIMIZATION SYSTEM
```

Restrictions propres aux sources de données des tables de requêtes matérialisées

Lorsque vous créez des tables de requêtes matérialisées, vous devez connaître les restrictions relatives aux sources de données spécifiques.

La présente rubrique décrit les restrictions relatives à la création de tables de requêtes matérialisées pour les sources de données suivantes :

- BioRS
- Fichiers structurés sous forme de table
- services Web
- XML

Restrictions de la recherche BioRS

L'encapsuleur BioRS requiert au moins un prédicat dans la clause WHERE. Vous devez créer une table de requêtes matérialisées qui satisfait les exigences de prédicat des encapsuleurs. Si vous ne spécifiez pas de prédicat, une régénération de la table de requêtes matérialisée échoue.

Restrictions relatives aux fichiers structurés en tableaux

Si vous définissez un pseudonyme pour un fichier structuré en tableaux avec l'option DOCUMENT, la table de requêtes matérialisée doit comporter un prédicat qui spécifie le chemin d'accès de ce fichier. Si vous ne spécifiez pas de prédicat, une régénération de la table de requêtes matérialisée échoue.

Restrictions relatives aux services Web

Vous ne pouvez créer une table de requêtes matérialisée que sur une vue à plat d'une hiérarchie de pseudonymes. Vous ne pouvez pas créer de table de requêtes matérialisée pour chaque pseudonyme d'une hiérarchie.

Restrictions relatives au langage XML

Vous ne pouvez pas créer de table matérialisée sur une table enfant.

Si vous définissez un pseudonyme pour une table XML avec l'option DOCUMENT, la table de requêtes matérialisée requiert un prédicat qui spécifie le chemin d'accès de ce fichier. Si vous ne spécifiez pas de prédicat, une régénération de la table de requêtes matérialisée échoue.

Restrictions sur l'utilisation des tables de requêtes matérialisées avec les pseudonymes

Les restrictions relatives aux tables de requêtes matérialisées qui font référence à des pseudonymes doivent être prises en considération lorsque vous optimisez votre système fédéré.

Contrôle d'accès basé sur les labels (LBAC) pour les objets de source de données

Les pseudonymes des objets de source de données utilisant le contrôle d'accès basé sur les labels ou Oracle Label Security ne peuvent pas être placés en mémoire cache et aucune table de requêtes matérialisée ne peut y être créée.

Tables de requêtes matérialisées gérées par le système

Le système fédéré ne prend pas en charge les tables de requêtes matérialisées qui font référence à des pseudonymes dans un environnement de base de données partitionnée.

L'encapsuleur fédéré doit être isolé pour que la fonction de réécriture fédérée achemine la requête à la MQT (table de requêtes matérialisées). Vous avez le choix entre créer l'encapsuleur comme FENCED ou le modifier à l'aide de l'instruction ALTER WRAPPER. Par exemple :

```
CREATE WRAPPER <wrapper name>
  LIBRARY <libname>
  OPTIONS (DB2_FENCED 'N');
```

```
ALTER WRAPPER <wrapper name> OPTIONS (SET DB2_FENCED 'Y');
```

Il vous faut définir les types de tables tenues à jour pour l'optimisation sur ALL ou USER à l'aide du paramètre de configuration de base de données DFT_MTTB_TYPES ou le registre spécial défini en cours émis avant le SQL.

Pour passer le paramètre de configuration sur la valeur USER, émettez :

```
update db cfg for <dbalias> using DFT_MTTB_TYPES USER
```

Pour utiliser le registre spécial défini en cours, émettez :

```
set current maintained table types for optimization ALL
```

Pour remédier à cette restriction, vous pouvez utiliser des tables de requêtes matérialisées gérées par l'utilisateur.

Par exemple, pour le pseudonyme non relationnel DEPART, vous pouvez exécuter les commandes ci-après afin de simuler une table de requêtes matérialisée gérée par le système.

```
SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION ALL;
```

```
CREATE TABLE AST1(C1, C2)
  AS (SELECT EMPNO, FIRSTNME FROM DEPART WHERE EMPNO>'000000')
  DATA INITIALLY DEFERRED REFRESH DEFERRED
  ENABLE QUERY OPTIMIZATION MAINTAINED BY USER;
```

```
SET INTEGRITY FOR AST1 ALL IMMEDIATE UNCHECKED;
```

```
INSERT INTO AST1 (SELECT EMPNO, FIRSTNME FROM DEPART WHERE EMPNO>'000000');
```

```
SET CURRENT REFRESH AGE ANY;
```

Il est possible de répondre à l'instruction SELECT suivante à l'aide de la table de requêtes matérialisée ci-dessus :

```
SELECT EMPNO, FIRSTNME FROM DEPART
  WHERE EMPNO > '000000' AND FIRSTNME LIKE 'AN%';
```

Chapitre 29. Tables de la mémoire cache

Les tables de mémoire cache permettent de stocker les données auxquelles vous accédez régulièrement mais dont le contenu change peu.

Une table de mémoire cache peut améliorer les performances des requêtes en stockant les données localement au lieu d'accéder directement aux données à partir de la source de données.

Vous pouvez mettre des données en mémoire cache à partir des sources de données suivantes :

- Famille DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

Une table de mémoire cache se compose des composants suivants :

- Un pseudonyme sur votre système de base de données fédérée. Le pseudonyme contient des définitions de colonne et un accès aux données identiques à ceux de la table de la source de données.
- Une ou plusieurs tables de requêtes matérialisées définies pour le pseudonyme. Le type de tables de requêtes matérialisées est `FEDERATED_TOOL` et permet de maintenir une table de requêtes matérialisée. Cette dernière contient généralement un sous-ensemble de données à utilisation intensive issues de la table de la source de données.
- Un planning de réplication pour chaque table de requêtes matérialisée. Ce planning maintient les tables de requêtes matérialisées locales à jour avec les tables de votre source de données. Définissez le planning de réplication.

L'exemple suivant illustre une table de mémoire cache.

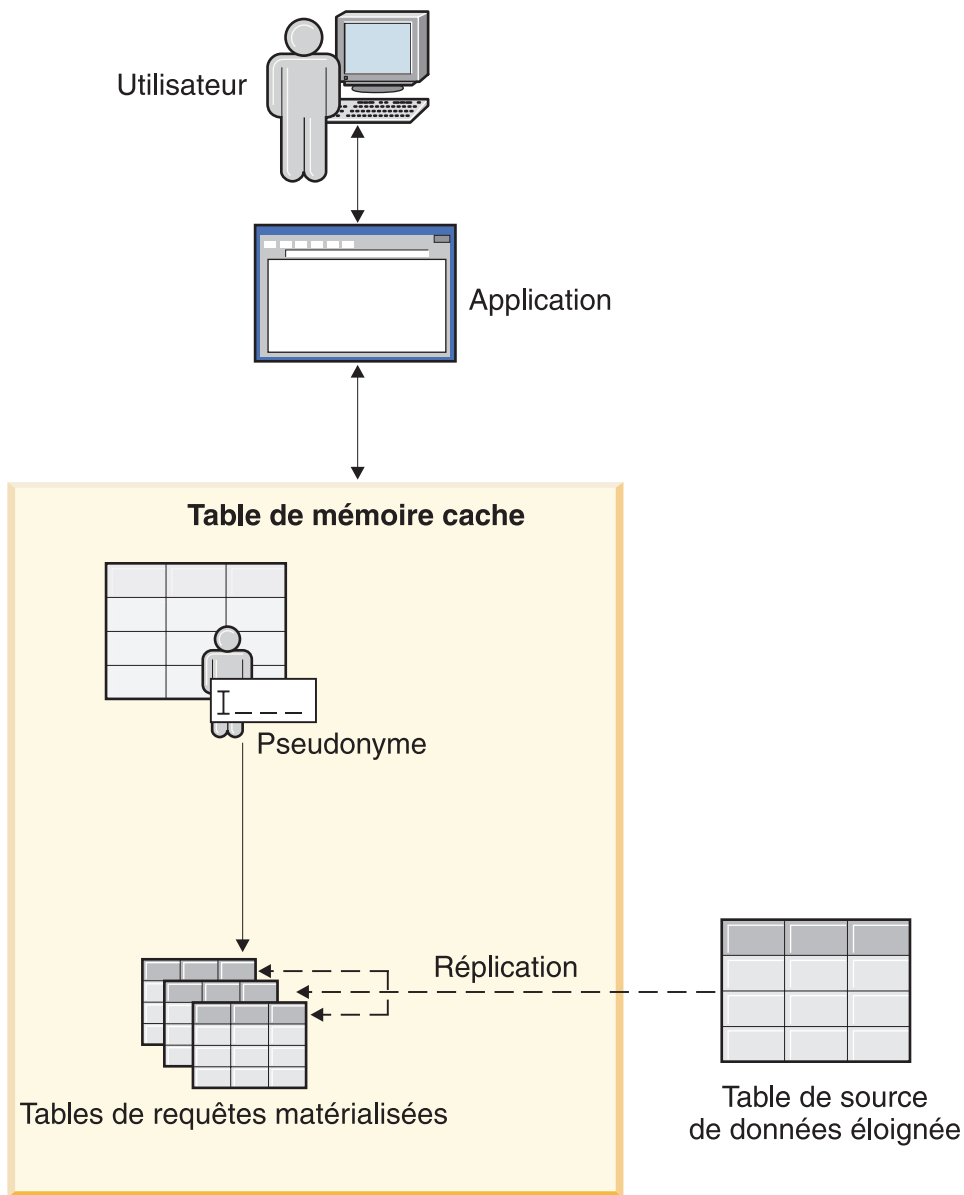


Figure 14. Table de mémoire cache.

La table de mémoire cache porte le même nom que le pseudonyme. Vous pouvez associer une table de mémoire cache à une seule table de source de données.

Lorsqu'une table de mémoire cache est activée, l'optimiseur de requêtes transmet les requêtes à la table de mémoire cache si les données demandées par la requête se trouvent dans la table de requêtes matérialisée.

Création de tables de mémoire cache

Utilisez un assistant dans le Centre de contrôle pour créer une table de mémoire cache. L'assistant crée le pseudonyme, la table de requêtes matérialisée et le planning de réplication qui sont requis pour la table de mémoire cache.

Avant de commencer

- Définissez le paramètre FEDERATED sur YES sur le serveur fédéré. Le paramètre FEDERATED est un paramètre de configuration du gestionnaire de base de données.
- Pour accéder aux sources de données Informix, installez et configurez le logiciel Informix Client SDK dans le serveur fédéré.
- Pour mettre en cache des données issues de DB2 Database pour les tables Linux, UNIX et Windows, configurez la base de données DB2 pour la consignation des archives.
- La base de données fédérée ou la base de données source doit se trouver sur l'ordinateur à partir duquel les tables de mémoire cache sont créées. Si la base de données fédérée ou la base de données source n'est pas locale, vous devez cataloguer les bases de données sur l'ordinateur local. Le nom de l'alias que vous utilisez lorsque vous cataloguez la base de données doit être identique au nom de la base de données.
- L'ID utilisateur dans le mappage utilisateur effectué entre les bases de données doit disposer d'un droit permettant de créer des tables dans la base de données source.

Procédure

Pour créer une table de mémoire cache, procédez comme suit :

1. Dans le Centre de contrôle, développez le dossier **Objets cache**.
2. Cliquez avec le bouton droit de la souris sur le dossier **Tables de mémoire cache**, puis cliquez sur **Créer**.
3. Exécutez les différentes étapes de l'assistant Table de mémoire cache pour créer une table de mémoire cache. Vous pouvez rapidement créer cette dernière en ne spécifiant que les valeurs des zones obligatoires et en utilisant les paramètres par défaut des zones restantes. Pour modifier les paramètres par défaut, procédez comme suit :
 - a. Dans la page Table de requêtes matérialisée, cliquez sur **Paramètres avancés de la table de requêtes matérialisée** pour modifier les paramètres par défaut ou pour sélectionner un sous-ensemble de colonnes de la table de requête matérialisée.
 - b. Dans la page Réplication, cliquez sur **Paramètres avancés** pour modifier les paramètres par défaut afin de répliquer les données issues de la table de source de données vers la table de requêtes matérialisée.

Dans certains cas, la mise en cache est désactivée lorsque vous utilisez l'assistant. Vous devez l'activer afin de lancer les programmes Capture et Apply pour la réplication.

L'assistant Table de mémoire cache crée une table de requêtes matérialisée lorsque vous créez la table de mémoire cache. Vous pouvez créer des tables de requêtes matérialisées supplémentaires pour stocker d'autres données issues de la même source de données.

Modification des paramètres pour les tables de requêtes matérialisées

Vous ne pouvez pas modifier directement les paramètres des tables de requêtes matérialisées. Vous devez utiliser des méthodes alternatives pour modifier les paramètres de réplication et de la table de requêtes matérialisée.

Procédure

Pour modifier les paramètres d'une table de requêtes matérialisée, procédez comme suit :

1. Utilisez la fenêtre Détails de la table de requêtes matérialisée pour afficher les paramètres de la table de requêtes matérialisée et le planning de réplication.
 - a. Dans le Centre de contrôle, développez le dossier **Objets cache**.
 - b. Cliquez avec le bouton droit de la souris sur la table de mémoire cache et cliquez sur **Propriétés**.
 - c. Sélectionnez la table de requêtes matérialisée et cliquez sur **Détails** pour afficher les paramètres actuels.
2. Si vous devez modifier les paramètres de réplication, utilisez le Centre de réplication. Vous ne pouvez pas modifier les paramètres de réplication de la table de requêtes matérialisées à partir de la fenêtre Détails de la table de requêtes matérialisée.
3. Si vous devez modifier les paramètres d'une table de requêtes matérialisée, supprimez-la et créez-en une autre. Par exemple, si vous devez ajouter une autre colonne à la table de requêtes matérialisée, supprimez cette table et créez-en une autre avec de nouveaux paramètres.

Ajout de tables de requêtes matérialisées à une table de mémoire cache

Vous pouvez créer des tables de requêtes matérialisées supplémentaires pour une même table de mémoire cache. Utilisez les tables de requêtes matérialisées supplémentaires pour stocker d'autres données issues d'une même table de source de données.

A propos de cette tâche

Lors de la création d'une table de mémoire cache, le serveur fédéré stocke localement les données de la source de données dans une table de requêtes matérialisée. Les critères que vous spécifiez dans l'assistant Table de mémoire cache déterminent les données à stocker dans la table de requêtes matérialisée.

Par exemple, vous disposez d'une table de requêtes matérialisée initiale qui contient des informations sur des clients situés en Asie. Vous pouvez créer une autre table de requêtes matérialisée qui contient des informations sur des clients situés en Amérique du Sud.

Procédure

Pour ajouter une table de requêtes matérialisée à une table de mémoire cache existante, procédez comme suit :

1. Dans le Centre de contrôle, développez les dossiers **Objets cache** et **Tables de mémoire cache**.

2. Cliquez avec le bouton droit de la souris sur la table de mémoire cache adéquate et cliquez sur **Propriétés**.
3. Cliquez sur **Ajouter**.
4. Exécutez les différentes étapes de l'assistant Table de mémoire cache pour créer une table de requêtes matérialisée supplémentaire.

Vous pouvez également accéder à la fenêtre Propriétés de la table de mémoire cache à partir d'un objet pseudonyme. Cliquez avec le bouton droit de la souris sur le pseudonyme et cliquez sur **Mettre en cache**.

Routage de requêtes vers des tables de mémoire cache

Vous pouvez router des requêtes vers la table de requêtes matérialisée pour la table de mémoire cache ou vers une source de données en acheminant les requêtes vers le pseudonyme.

Avant de commencer

- Vous devez disposer d'une table de mémoire cache pour le pseudonyme interrogé.
- Activez la mise en cache de la table de requêtes matérialisée.
- Définissez les paramètres de configuration de la base de données suivants :
 - Type de table matérialisée pour l'optimisation (DFT_MTTB_TYPES)
 - Classe d'optimisation des requêtes (DFT_QUERYOPT)

Utilisez le bloc-notes **Configuration de la base de données** dans le Centre de contrôle ou la ligne de commande pour définir ces paramètres.

Procédure

Pour modifier le routage des requêtes, procédez comme suit :

1. Dans le Centre de contrôle, développez les dossiers **Objets cache** et **Tables de mémoire cache**.
2. Cliquez avec le bouton droit de la souris sur la table de mémoire cache adéquate et cliquez sur **Propriétés**.
3. Sélectionnez la table de requêtes matérialisée et cliquez sur **Vérifier l'état**.
4. Définissez l'emplacement vers lequel les requêtes doivent router les requêtes :
 - Sélectionnez **Router vers la table de requêtes matérialisée**. Toutes les requêtes de la source de données sont envoyées vers la table de requêtes matérialisée. Si les données figurant dans la table de requêtes matérialisée ne peuvent pas satisfaire la requête, cette dernière est alors routée vers la source de données à l'aide du pseudonyme.
 - Sélectionnez **Router vers le pseudonyme**. Toutes les requêtes sont envoyées vers la source de données à l'aide du pseudonyme. La table de requêtes matérialisée devient alors une table de base de données classique. La réplication vers la table de requêtes matérialisée se poursuit jusqu'à ce que vous désactiviez la mise en cache.
5. Pour modifier instantanément le routage, cochez la case **Supprimer toutes les instructions SQL dynamiques mises en cache**. Les instructions SQL dynamiques qui se trouvent actuellement dans la mémoire cache du module sont supprimées.
6. Cliquez sur **OK**.

Vous pouvez également accéder à la fenêtre Propriétés de la table de mémoire cache à partir d'un objet pseudonyme. Cliquez avec le bouton droit de la souris sur le pseudonyme et cliquez sur **Mettre en cache**.

Activation et désactivation des paramètres du cache de réplication

Vous pouvez lancer et arrêter la réplication de données de la table de requêtes matérialisée en modifiant les paramètres de la mémoire cache.

Avant de commencer

Pour les sources de données DB2 Database pour Linux, UNIX et Windows, définissez le type de consignation de la base de données par rapport à la consignation des archives.

A propos de cette tâche

Lors de l'activation de la mise en cache, les programmes Capture et Apply sont démarrés s'ils ne sont pas déjà en cours d'exécution. Le membre d'un ensemble d'abonnements est également activé. Son activation indique au programme Apply de conserver les données dans la table de requêtes matérialisée synchronisée avec celles figurant dans la table de source de données.

Lors de la désactivation de la mise en cache, le membre d'un ensemble d'abonnements est désactivé. Les données issues de la source de données ne sont pas répliquées vers la table de requêtes matérialisée.

Important : Si vous ne modifiez pas le paramètre de routage, vos requêtes sont routées vers la table de requêtes matérialisée même si les données ne sont pas répliquées vers cette table.

Procédure

Pour activer ou désactiver les paramètres de mise en cache de la réplication d'une table de requêtes matérialisée, procédez comme suit :

1. Dans le Centre de contrôle, développez les dossiers **Objets cache** et **Tables de mémoire cache**.
2. Cliquez avec le bouton droit de la souris sur la table de mémoire cache adéquate et cliquez sur **Propriétés**.
3. Sélectionnez la table de requêtes matérialisée et cliquez sur **Vérifier l'état**.
4. Sélectionnez **Activer la mise en cache** ou **Désactiver la mise en cache**. Pour afficher les commandes d'activation ou de désactivation, cliquez sur **Afficher les instructions**.
5. Cliquez sur **OK** .

Vous pouvez également accéder à la fenêtre **Propriétés de la table de mémoire cache** à partir d'un objet pseudonyme. Cliquez avec le bouton droit de la souris sur le pseudonyme et cliquez sur **Mettre en cache**.

Suppression de tables de requêtes matérialisées d'une table de mémoire cache

Si vous ne voulez plus stocker de données localement dans une table de requêtes matérialisée, vous pouvez supprimer cette table de la table de mémoire cache.

A propos de cette tâche

Si une table de mémoire cache dispose d'une seule table de requêtes matérialisée, la suppression de cette dernière entraîne la suppression de la table de mémoire cache.

Pour vous assurer que la table de requêtes matérialisée a été entièrement supprimée de votre système, utilisez le Centre de contrôle pour la supprimer d'une table de mémoire cache.

Procédure

Pour supprimer une table de requêtes matérialisée d'une table de mémoire cache, procédez comme suit :

1. Dans le Centre de contrôle, développez les dossiers **Objets cache** et **Tables de mémoire cache** dans l'arborescence d'objets.
2. Cliquez avec le bouton droit de la souris sur la table de mémoire cache adéquate et cliquez sur **Propriétés**.
3. Sélectionnez la table de requêtes matérialisée adéquate et cliquez sur **Supprimer**.

Suppression de tables de mémoire cache

Si vous ne voulez plus stocker de données localement dans une table de mémoire cache, vous pouvez la supprimer.

A propos de cette tâche

Lors de la suppression d'une table de mémoire cache, le serveur fédéré exécute les actions suivantes :

- Supprime les tables de requêtes matérialisées qui sont destinées à la table de mémoire cache.
- Supprime le planning de réplication des sources de données et des tables de requêtes matérialisées.
- Supprime le pseudonyme de la source de données, si le pseudonyme a été créé en même temps que la table de mémoire cache. Si vous avez utilisé un pseudonyme existant lors de la création de la table de mémoire cache, le serveur fédéré ne supprime pas ce pseudonyme.

Procédure

Pour supprimer une table de mémoire cache, procédez comme suit :

1. Dans le Centre de contrôle, développez les dossiers **Objets cache** et **Tables de mémoire cache** dans l'arborescence d'objets.
2. Cliquez avec le bouton droit de la souris sur la table de mémoire cache adéquate et cliquez sur **Supprimer**.

Chapitre 30. Sécurité pour les serveurs fédérés

Le serveur fédéré prend en charge Secure Socket Layer (SSL) pour le chiffrement de données, ainsi que les proxys HTTP et SOCKS pour les sources de données spécifiques.

Chiffrement

Le chiffrement offre un niveau de sécurité supérieur à celui offert par un simple nom et mot de passe. Le standard Internet de chiffrement entre les noeuds finaux est le protocole SSL (Secure Socket Layer) et TLS (Transport Layer Security). SSL utilise des certificats signés pour garantir la sécurité des communications. Un *certificat* est un document numérique qui garantit l'identité d'un utilisateur ou d'un serveur. Un certificat est signé par une autorité de certification, telle que VeriSign ou peut être auto-signé. Chaque partenaire de la communication détermine ou non s'il accepte un certificat particulier comme étant authentique.

Chaque partenaire de la communication dispose d'un espace de stockage de certificats ou d'un fichier de clés. Le fichier de clés détient les certificats que le partenaire accepte des autres, ainsi que les certificats qui le représentent lui-même. Chaque noeud final détermine s'il enverra ou non un certificat lors de l'ouverture des communications et s'il acceptera ou non les communications provenant d'un partenaire qui ne fournit pas de certificat.

En ce qui concerne les encapsuleurs et les fonctions, les dispositifs SSL suivants sont pertinents :

- Identification côté serveur d'un certificat à envoyer
- Vérification côté serveur d'un certificat client
- Vérification côté client d'un certificat serveur
- Identification côté client d'un certificat à envoyer
- Type de prise en charge serveur et client, emplacement et accès au fichier de clés local

L'interaction entre SSL et un proxy dépend du type de proxy. En général, les communications SSL sont mises en tunnel ou relayées via un proxy. La session de proxy est établie en texte clair.

IBM Global Security Kit (GSKit) offre des services de chiffrement pour les encapsuleurs et les fonctions définies par l'utilisateur.

Proxys

Avec pour objectif de contrecarrer les différents types d'attaque sur Internet, de nombreuses entreprises implémentent un pare-feu. Un *pare-feu* est une configuration réseau qui repose sur une installation matérielle et logicielle généralement située à une frontière de communication, par exemple entre un Intranet d'entreprise et Internet. Le pare-feu agit comme un contrôleur d'accès afin de réguler le trafic à la frontière. Dans la plupart des cas, le pare-feu empêche les communications indésirables de traverser la frontière ; cependant, occasionnellement, il peut arriver que le pare-feu bloque des communications légitimes.

Pour s'assurer que toutes les communications légitimes passent le pare-feu, vous pouvez implémenter un proxy. Un *proxy* est un programme serveur autorisé à communiquer via le pare-feu. Ensuite, lorsqu'un programme utilisateur a besoin de se connecter à un serveur distant, le programme utilisateur effectue une demande auprès du proxy qui effectue la connexion au serveur distant. Une fois la connexion établie, le proxy contrôle le trafic entre le programme utilisateur et le serveur distant. Ce processus permet au programme utilisateur de traverser le pare-feu et garantit la sécurité : le serveur distant ne connaît que l'adresse du proxy et pas l'adresse du programme utilisateur.

Les serveurs de proxy SOCKS et HTTP contrôlent le trafic entre les programmes utilisateur et les serveurs distants. Un proxy SOCKS fonctionne au niveau de la couche de transport et relaie les messages de transport arbitraires (TCP ou UDP) entre deux adresses. Le serveur fédéré prend en charge SOCKS4 et SOCKS5. SOCKS4 qui prend en charge IPv4 ne prend pas en charge l'authentification d'utilisateur. C'est pourquoi, n'importe qui peut communiquer via un proxy SOCKS4 sans avoir à fournir de données d'identification de l'utilisateur. SOCKS5 qui prend en charge IPv6 prend en charge différents modes d'authentification d'utilisateur. L'IETF (Internet Engineering Task Force) a approuvé SOCKS5 en tant que standard. Pour plus d'informations, visitez la page Web www.ietf.org et consultez RFC1928, RFC1929 et RFC1961.

Pour utiliser un proxy SOCKS, la couche transport doit être configurée à l'avance en vue d'utiliser un proxy. Après l'ouverture d'une connexion vers le proxy, la couche transport demande au proxy d'ouvrir une connexion vers le serveur distant. Si la configuration nécessite une authentification, il est possible que le proxy SOCKS demande au programme de fournir un ID et un mot de passe avant d'ouvrir la connexion vers le serveur distant.

Un proxy HTTP fonctionne avec le protocole HTTP qui est un protocole de couche application. Après l'ouverture d'une connexion TCP/IP vers le proxy, le programme utilisateur envoie une requête qui comprend le nom du serveur distant. Ensuite, le programme utilisateur continue à soumettre des requêtes via le serveur de proxy. Ce processus diffère légèrement si le serveur distant nécessite une authentification. Si le serveur distant nécessite une authentification, le serveur de proxy envoie un message de réponse qui comprend l'en-tête de demande d'authentification de proxy. Cet en-tête comprend des informations sur le type d'authentification à réaliser. Le programme utilisateur soumet ensuite à nouveau la requête et inclut un en-tête d'authentification de proxy qui contient la réponse à la question d'authentification.

Chapitre 31. Contextes accrédités fédérés et connexions accréditées

Améliorez les performances du système et réduisez ou réduisez complètement l'utilisation et la maintenance des mappages utilisateur.

Un *contexte accrédité* est un objet de base de données DB2 qui définit une relation d'accréditation entre un client et une source de données, par exemple, entre un serveur d'application et un serveur fédéré, ou entre un serveur fédéré et un serveur de base de données distant. Pour définir une relation accréditée, le contexte accrédité spécifie les *attributs d'accréditation*. Trois types d'attributs d'accréditation existent :

- L'ID d'autorisation du système qui effectue la requête initiale de connexion à la base de données
- L'adresse IP ou le nom de domaine à partir duquel la connexion est réalisée
- La configuration du chiffrement pour les communications de données entre le serveur de base de données et le client de base de données

Une *connexion accréditée* est établie lorsque tous les attributs d'une requête de connexion correspondent aux attributs d'accréditation spécifiés dans tout objet de contexte accrédité défini sur le serveur. Après l'établissement d'une connexion accréditée explicite, les utilisateurs peuvent être basculés sur la même connexion physique, avec ou sans authentification. De plus, les utilisateurs peuvent se voir accorder des rôles qui spécifient les privilèges qui ne doivent être utilisés que dans la connexion accréditée.

Cet exemple crée un objet de contexte accrédité pour BOSS :

```
CREATE TRUSTED CONTEXT MYCTX
  BASED UPON CONNECTION USING SYSTEM AUTHID BOSS
  ATTRIBUTES (ADDRESS '9.26.111.111')
  WITH USE FOR MARY WITH AUTHENTICATION ROLE MANAGER,
  PUBLIC WITHOUT AUTHENTICATION
  DEFAULT ROLE AUDITOR
  ENABLE
```

Dans cet exemple, seul BOSS peut initier une connexion accréditée à partir de l'adresse IP 9.26.111.111. Mary peut réutiliser la connexion mais elle doit d'abord s'authentifier. Puis elle obtient un rôle supplémentaire de MANAGER, qui spécifie les privilèges que Mary peut utiliser dans cette connexion accréditée. Les autres utilisateurs, spécifiés en tant que PUBLIC, peuvent réutiliser la connexion, et ils n'ont pas besoin de s'authentifier. Ces autres utilisateurs obtiennent le rôle supplémentaire d'AUDITOR, qui spécifie les privilèges qu'ils peuvent utiliser dans cette connexion accréditée. Ces privilèges supplémentaires ne sont à la disposition des utilisateurs que lorsqu'ils sont des utilisateurs actifs de la connexion accréditée.

Une connexion accréditée est soit explicite soit implicite. Le type de connexion détermine si la connexion peut être réutilisée et si les utilisateurs peuvent obtenir des rôles supplémentaires.

Une *connexion accréditée implicite* est établie lorsque tous les attributs d'une requête de connexion correspondent aux attributs d'accréditation spécifiés dans tout objet de contexte accrédité défini sur le serveur. Dès qu'une connexion accréditée implicite est établie, seul l'émetteur de la connexion accréditée peut hériter des

rôles qui ne sont autrement pas à sa disposition. Une connexion accréditée implicite ne peut pas être réutilisée par d'autres utilisateurs.

Une *connexion accréditée implicite* est établie lorsqu'une application utilise une API pour demander une connexion accréditée. Si les attributs de connexion correspondent aux attributs d'accréditation d'un contexte accrédité, une connexion accréditée est établie. Sinon, une connexion normale est établie. Après qu'une connexion soit établie, d'autres utilisateurs peuvent réutiliser la connexion ; et l'émetteur de la connexion ainsi que les utilisateurs de la connexion héritent de rôles supplémentaires qui ne sont autrement pas à leur disposition.

Avantages des connexions accréditées fédérées

Dans un modèle d'application multiniveaux, les connexions accréditées fédérées réutilisent une unique connexion physique pour propager l'identité réelle de chaque utilisateur à travers les niveaux du serveur de base de données.

Pour comprendre les avantages des connexions accréditées fédérées, envisagez les problèmes inhérents à un modèle d'application multiniveaux type. Un modèle d'application multiniveaux est composé d'utilisateurs entreprise (Niveau 1) qui échangent avec une application qui s'exécute sur un serveur d'application (Niveau 2), qui achemine tous les accès de base de données via le serveur fédéré (Niveau 3), qui gère les communications avec les divers serveurs de bases de données (Niveau 4). Dans ce modèle, le serveur d'application authentifie les utilisateurs et gère les échanges avec le serveur fédéré. Le serveur fédéré traduit les demandes utilisateurs en formats spécifiques aux sources de données, établit des connexions avec les sources de données distantes, et leur envoie des requêtes.

Ce modèle utilise l'ID et le mot de passe du serveur d'application pour créer une connexion au serveur de base de données ; le serveur fédéré transmet simplement l'ID et le mot de passe du serveur d'application au serveur de base de données. Le serveur de base de données utilise les privilèges de base de données associés à cet ID pour autoriser et auditer toutes les transactions exécutées par le serveur d'application, y compris toutes les transactions que le serveur d'application exécute au nom des utilisateurs de l'entreprise.

L'utilisation de l'ID du serveur d'application présente les problèmes suivants :

- L'identité réelle de l'utilisateur qui exécute une transaction est inconnue car le serveur d'application exécute toutes les transactions.
- Les utilisateurs ne peuvent être tenus pour responsables des transactions car elles ne peuvent être auditées.
- Le principe du moindre privilège est enfreint car l'ID de serveur d'application nécessite une version élaborée de tous les privilèges requis par tous les utilisateurs.
- Les données sont vulnérables si l'ID de serveur d'application est compromis.

Les connexions accréditées fédérées traitent ces problèmes et apportent ces avantages qui peuvent renforcer la sécurité et les performances du système :

L'identité de l'utilisateur est connue

Dans la mesure où les utilisateurs sont basculés sur la connexion, vous connaissez l'identité réelle des utilisateurs qui accèdent à la base de données.

Les utilisateurs sont tenus pour responsables

Les journaux d'audit pour la base de données fédérée et pour les bases de données de sources de données distantes identifient les transactions que le serveur d'application exécute à ses propres fins et les transactions que chaque utilisateur exécute. Par conséquent, vous pouvez tenir les utilisateurs spécifiques comme responsables de transactions spécifiques.

Les privilèges sont limités

Lorsque vous créez un contexte accredité, vous pouvez accorder un rôle de base de données par défaut à tous les utilisateurs et accorder des rôles spécifiques aux utilisateurs spécifiques. Seules les connexions de base de données accréditées qui correspondent à la définition de ce contexte accredité peuvent bénéficier des privilèges associés à ce rôle.

Les données sont moins vulnérables

Dans un système qui utilise les connexions accréditées fédérées, l'ID du serveur d'application ne nécessite pas de version élaborée de tous les privilèges dont les utilisateurs ont besoin. Par conséquent, si l'ID du serveur d'application est compromis, les données sont moins vulnérables que lorsque l'ID a une version élaborée de tous les privilèges dont les utilisateurs ont besoin.

La maintenance administrative est minimisée

La nécessité de créer et de maintenir les mappages utilisateurs est nettement réduite.

Les performances sont améliorées

Après l'établissement d'une connexion accréditée explicite, le serveur fédéré peut basculer l'ID de l'utilisateur en cours sur la connexion à un autre ID utilisateur, en exigeant ou non l'authentification de l'utilisateur. La réutilisation de la même connexion physique pour différents utilisateurs peut améliorer les performances.

Types de connexions accréditées fédérées

Les connexions accréditées fédérées sont soit des connexions accréditées de bout en bout soit des connexions accréditées sortantes. Le type de connexion effectué est fait en fonction de la manière dont vous configurez le système et si la demande de connexion entrante est ou non accréditée.

Une configuration fédérée type est mutliniveaux ; c'est-à-dire qu'elle comprend un serveur d'application, un serveur fédéré, et un serveur de source de données distant. Dans cette configuration, le serveur fédéré reçoit des demandes de connexion entrante du serveur d'application et fait des demandes de connexion sortantes au serveur de source de données distante.

Connexions fédérées accréditées de bout en bout

Une connexion fédérée accréditée de bout en bout permet une réutilisation de la connexion et une vérification de l'identité pour les connexions entrantes et sortantes. Par exemple, lorsqu'une connexion entrante au serveur fédéré est accréditée, que ce soit explicitement ou implicitement, le serveur fédéré demande automatiquement une connexion accréditée sortante. Si la source de données fournit une fonction de vérification d'identité, une connexion sortante accréditée est effectuée ; et l'identité de l'utilisateur est propagée dans la source de données distante. Les connexions entrantes et sortantes sont réutilisées chaque fois qu'un autre utilisateur demande à réutiliser la connexion accréditée, et que l'identité de ce nouvel utilisateur est propagée dans le système.

Pour les sources de données qui n'ont pas de fonction de vérification d'identité, le serveur fédéré fournit une vérification d'identité de bout en bout mais ne fournit pas de réutilisation de la connexion. Dans ces cas, chaque fois que l'utilisateur est basculé, le serveur fédéré ferme la connexion sortante pour l'utilisateur précédent et crée une nouvelle connexion pour le nouvel utilisateur. Ce faisant, l'identité de l'utilisateur est propagée dans le système, même si la connexion sortante n'est pas réutilisée.

connexions fédérées accréditées sortantes

Une connexion fédérée accréditée sortante utilise la capacité des sources de données pour réutiliser les connexions accréditées sans authentification pour éliminer la nécessité de stocker les mots de passe de sources de données dans les mappages utilisateur. Si la connexion entrante est accréditée, le serveur fédéré demande automatiquement une connexion sortante accréditée qui est réutilisée sans authentification. Pour les connexions entrantes non accréditées, les connexions accréditées fédérées sortantes fournissent la capacité de réutiliser les connexions sans qu'il soit besoin que les utilisateurs s'authentifient.

Dans cette configuration, vous utilisez l'option `FED_PROXY_USER` dans la définition du serveur afin de spécifier l'ID d'autorisation qui établit initialement la connexion sortante. L'ID d'autorisation que vous spécifiez doit avoir un mappage utilisateur comprenant à la fois les options `REMOTE_AUTHID` et `REMOTE_PASSWORD`.

Selon la manière dont vous configurez le contexte accrédité sur le serveur de source de données distant, vous pouvez réduire le nombre de mappages utilisateurs dans le catalogue de base de données à seulement quelques uns. Par exemple, si vous autorisez `PUBLIC` à se connecter sans authentification, alors seul l'utilisateur de proxy fédéré nécessite un mappage utilisateur. Cependant, si le contexte accrédité fédéré sur la source de données distante spécifie que certains utilisateurs n'utilisent pas le même ID utilisateur sur le serveur fédéré et sur la source de données distante, vous devez créer ou modifier le mappage utilisateur de l'utilisateur de façon à ce qu'il utilise uniquement l'option `REMOTE_PASSWORD`, ou qu'il utilise l'option `REMOTE_AUTHID` et l'option `REMOTE_PASSWORD` ; et vous devez définir l'option `USE_TRUSTED_CONTEXT` sur 'Y'.

Dans la mesure où il y a des tâches de configuration et de maintenance supplémentaires significatives affiliées à la connexion accréditée fédérée sortante, concevez le système fédéré afin qu'il implémente des connexions accréditées de bout en bout chaque fois que cela est possible. Puis utilisez les connexions accréditées fédérées sortantes uniquement lorsque cela est absolument nécessaire.

API pour les connexions accréditées fédérées

Les API que vous utilisez pour requêter et réutiliser une connexion accréditée dépendent du type d'application.

Pour requêter une connexion accréditée fédérée, l'application doit spécifier ces API.

applications CLI/ODBC

Utilisez `SQLSetConnectAttr` avec l'attribut `SQL_ATTR_USE_TRUSTED_CONTEXT` pour indiquer si le client demande une connexion accréditée. Puis émettez une `SQLConnect`.

applications XA CLI/ODBC

Dans une requête de chaîne xa_open, définissez l'attribut TCTX sur Vrai pour demander une connexion accréditée.

Applications Java

Utilisez getDB2TrustedPooledConnection ou getDB2TrustedXAConnection pour demander une connexion accréditée.

Pour réutiliser la connexion pour un autre utilisateur, en demandant ou non une authentification, l'application doit spécifier les API suivantes :

applications CLI/ODBC

Utilisez SQLSetConnectAttr pour définir le SQL_ATTR_TRUSTED_CONTEXT_USERID et le SQL_ATTR_TRUSTED_CONTEXT_PASSWORD afin de spécifier l'ID utilisateur et le mot de passe de l'utilisateur auquel la connexion est commutée.

applications XA CLI/ODBC

Utilisez SQLSetConnectAttr pour définir le SQL_ATTR_TRUSTED_CONTEXT_USERID et le SQL_ATTR_TRUSTED_CONTEXT_PASSWORD afin de spécifier l'ID utilisateur et le mot de passe de l'utilisateur auquel la connexion est commutée.

Applications Java

Utilisez getDB2Connection et reuseDB2Connection.

Scénarios pour l'implémentation de connexions accréditées fédérées

Chaque système fédéré est unique ; c'est pourquoi il n'y a pas d'instructions étape par étape pour l'implémentation des contextes accrédités fédérés. Utilisez ces scénarios pour comprendre parfaitement les connexions accréditées fédérées ; puis planifiez et implémentez votre propre solution.

Chaque scénario utilise le même système fédéré multiniveaux qui comprend une application s'exécutant sur un serveur d'application, un serveur fédéré, et un serveur de base de données DB2 distant. Les scénarios illustrent la configuration des sources distantes et des serveurs fédérés pour utiliser les connexions accréditées fédérées.

Le premier scénario qui ne nécessite pas de mappage utilisateur, est le plus simple à implémenter et à maintenir. En fait, c'est le scénario recommandé lorsque vous implémentez les connexions accréditées sur un nouveau système.

Le deuxième scénario illustre l'implémentation des connexions accréditées dans un système comprenant des mappages utilisateur. Le troisième scénario illustre l'utilisation des connexions accréditées fédérées pour éliminer les mappages utilisateur. Gardez à l'esprit que ces scénarios sont plus complexes et exigent plus d'efforts pour la configuration et la maintenance que le premier scénario.

Scénario : connexions accréditées fédérées de bout en bout, sans mappages utilisateur

Les mappages utilisateurs exigent une importante maintenance administrative. Ce scénario illustre comment configurer les connexions accréditées fédérées de façon à ce que le système fédéré ne nécessite pas de mappages utilisateur.

Exigences d'ID utilisateur et de mot de passe

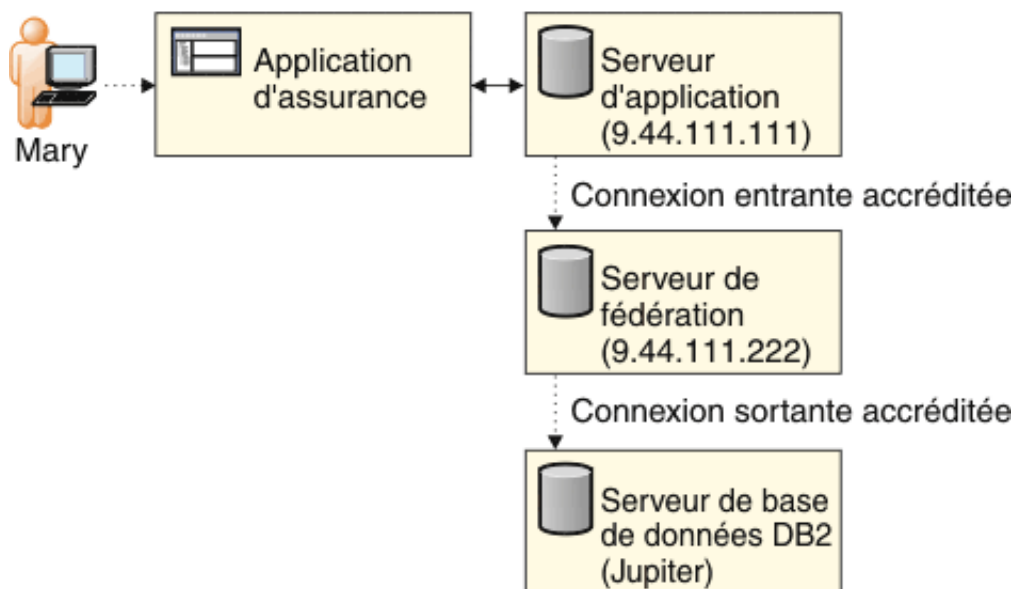
Pour définir des contextes accrédités sans utiliser de mappages utilisateur, les ID utilisateur et mots de passe doivent se conformer aux exigences suivantes :

- L'ID utilisateur et le mot de passe pour l'émetteur de la connexion doivent être à la disposition du serveur fédéré, et l'ID utilisateur et mot de passe de l'émetteur de la connexion doivent être identiques sur le serveur fédéré et sur la source de données. Les droits d'accès sont identifiés dans une instruction CONNECT que l'émetteur de la connexion émet ou dans un appel d'API effectué par l'application. Si l'instruction CONNECT est utilisée, une connexion accréditée implicite est créée et la connexion ne peut pas être réutilisée. Si l'application effectue un appel d'API et demande explicitement une connexion accréditée, une connexion accréditée explicite est créée, et la connexion peut être réutilisée.
- Si le contexte accrédité distant permet une réutilisation de la connexion sans authentification, les nouveaux utilisateurs de la connexion doivent avoir le même ID utilisateur sur le serveur fédéré et sur la source de données distante.
- Si le contexte accrédité distant permet une réutilisation de la connexion avec authentification, les nouveaux utilisateurs de la connexion doivent avoir le même name utilisateur sur le serveur fédéré et sur la source de données distante.

Le scénario

La figure suivante illustre un système fédéré multiniveau type qui est configuré pour utiliser les connexions accréditées fédérées de bout en bout. Bien que ce scénario inclue un serveur d'application, tout client de base de données peut établir une connexion accréditée.

Emetteur de la connexion : BOSS



Ce scénario a deux utilisateurs, dont aucune ne nécessite un mappage utilisateur :

- Le BOSS a le même ID utilisateur et mot de passe sur le serveur fédéré et sur la source de données. Par conséquent, le BOSS ne nécessite pas de mappage utilisateur.
- Mary utilise le même ID utilisateur sur le serveur fédéré et sur la source de données, et le contexte fédéré lui permet une connexion sans authentification. Par conséquent, elle ne nécessite pas de mappage utilisateur.

Ce scénario comprend trois serveurs :

- Le serveur d'application qui héberge l'application d'assurance et possède l'adresse IP 9.44.111.111.
- Le serveur de fédération, qui a l'adresse IP 9.44.111.222.
- Le serveur de base de données DB2 distant, qui est catalogué sur le serveur fédéré en tant que JUPITER.

Les étapes suivantes décrivent la configuration de ce scénario.

Remarque : Dans les commandes, les noms d'objet qui sont variables s'affichent en italique. Lorsque vous implémentez des contextes accrédités, spécifiez les noms de variables qui s'appliquent à votre configuration système spécifique.

1. Sur le serveur de base de données DB2 distant, créez cet objet de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_DB2_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Ce contexte accrédité spécifie que le BOSS est l'émetteur de la connexion accréditée et que la demande de connexion doit provenir de l'adresse IP 9.44.111.222, qui identifie le serveur de fédération. Le contexte accrédité spécifie WITH USE FOR PUBLIC WITHOUT AUTHENTICATION. Par conséquent, une fois la connexion accréditée établie, tout utilisateur valide de la source de données distante peut réutiliser la connexion en fournissant uniquement un ID utilisateur.

2. Sur le serveur de fédération, créez cet objet de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_WFS_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.111')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Ce contexte accrédité spécifie que le BOSS est l'émetteur de la connexion accréditée et que la demande de connexion doit provenir de l'adresse IP 9.44.111.111, qui identifie le serveur d'application. Par conséquent, une fois la connexion accréditée établie, tout utilisateur valide du serveur fédéré peut réutiliser la connexion en fournissant uniquement un ID utilisateur.

3. Sur le serveur de fédération, créez cette définition de serveur :

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb', ...);
```

Cette définition de serveur contient les informations dont le serveur fédéré a besoin pour se connecter à la base de données DB2 distante dénommée remotedb.

Le scénario, étape par étape

Voici une brève description étape par étape de la manière dont les connexions accréditées sont effectuées, et de la manière dont les utilisateurs sont basculés dans ce scénario. Le code du scénario comprend des commentaires décrivant la manière dont l'application exécute ces tâches.

1. Le serveur d'application demande une connexion entrante accréditée pour le BOSS.
2. le BOSS exécute une tâche, et le serveur fédéré établit une connexion accréditée sortante explicite pour le BOSS. L'ID utilisateur BOSS est propagé depuis le serveur d'application via le serveur de fédération vers le serveur de base de données DB2, où les actions effectuées par le BOSS peuvent être auditées.
3. Mary se connecte dans l'application d'assurance qui se trouve sur son ordinateur portable. Le serveur d'application bascule la connexion entrante sur le serveur fédéré à partir du BOSS vers Mary.
4. Mary exécute une tâche dans l'application.
5. Le serveur de fédération bascule la connexion sortante du BOSS vers Mary, et son ID est propagé via le serveur de fédération au serveur DB2, où les actions effectuées par Mary peuvent être auditées.

Code échantillon pour les scénarios de connexion accréditée fédérée de bout en bout

Ce code échantillon illustre comment utiliser les API dans une application qui utilise les connexions accréditées fédérées de bout en bout.

Une application doit utiliser les API pour demander une connexion entrante accréditée explicite et pour basculer l'utilisateur sur la connexion. Ce code échantillon illustre les parties de l'application qui exécutent ces tâches. Pour les deux scénarios de contexte accrédité de bout en bout, l'application est identique.

Cet extrait d'une application utilise les API d'interface de ligne de commande. Les API sont également disponibles pour les applications qui utilisent Java ou XA ODBC/CLI.

```
//Définissez l'attribut de connexion accréditée.
SQLSetConnectAttr(h1, SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER);

//Etablissez une connexion entrante accréditée pour le BOSS.
SQLConnect(h1, "testdb", SQL_NTS, "BOSS", SQL_NTS, "*****", SQL_NTS);

//Etablissez une connexion sortante accréditée pour le BOSS.
Exécutez quelques tâches sous l'ID utilisateur BOSS.
SQLExecDirect(hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS);
...
//Validez le travail.
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);

//A la limite de la transaction, basculez l'ID utilisateur entrant sur la connexion
accréditée à Mary.
SQLSetConnectAttr(h1, SQL_ATTR_TRUSTED_CONTEXT_USERID, "Mary", SQL_IS_POINTER);

//Basculez l'ID utilisateur sortant sur la connexion accréditée à Mary.
Exécutez quelques tâches sous l'ID utilisateur Mary.
SQLExecDirect(*hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS)

...
//Validez le travail.
```

```
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);  
//Déconnectez-vous de la base de données.  
SQLDisconnect(h1);
```

Scénario : connexions accréditées fédérées de bout en bout, avec mappages utilisateur

Les mappages utilisateurs sont requis lorsque les utilisateurs n'ont pas le même ID utilisateur et mot de passe sur le serveur fédéré et sur la source de données distante. Pour ce scénario, vous créez des mappages utilisateurs et configurez des contextes accrédités.

Exigences de mappage utilisateur

Le serveur fédéré reçoit des demandes de connexion entrante et fait des demandes de connexion sortantes à une source de données distante. Lorsque les utilisateurs ont le même ID utilisateur et mot de passe sur le serveur fédéré et sur le serveur de base de données DB2 distant, les mappages utilisateurs ne sont pas nécessaires. Cependant, lorsque les droits d'accès ne correspondent pas, un mappage utilisateur est nécessaire. Le mappage utilisateur mappe l'ID utilisateur de l'utilisateur sur le serveur fédéré avec l'ID utilisateur de l'utilisateur, et avec le mot de passe de l'utilisateur, si un mot de passe est spécifié, sur le serveur de base de données distant.

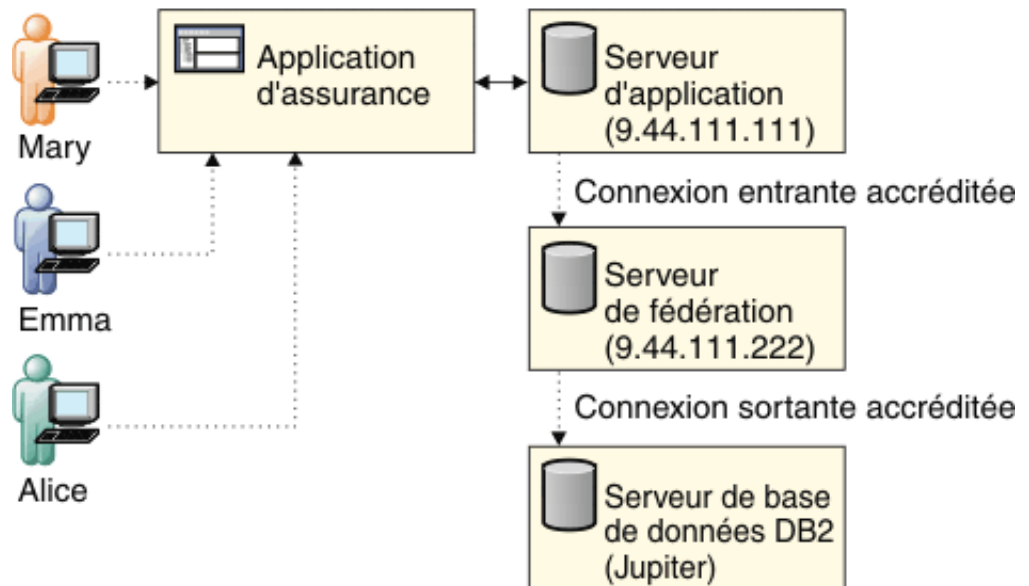
Dans un système fédéré qui utilise des contextes accrédités de bout en bout, les utilisateurs dont les noms et mots de passe sur le serveur fédéré et sur le serveur de base de données DB2 ne correspondent pas nécessitent un *mappage utilisateur accrédité*. Un mappage utilisateur accrédité spécifie que l'utilisateur a la permission d'utiliser un contexte accrédité. Pour créer un mappage utilisateur accrédité ou modifier un mappage utilisateur existant, vous définissez l'option de mappage utilisateur `USE_TRUSTED_CONTEXT` sur 'Y'.

La personne qui peut créer et modifier un mappage utilisateur accrédité est rigoureusement contrôlée. Seul un utilisateur qui a un droit d'accès SECADM peut créer ou supprimer un mappage utilisateur accrédité ou modifier un mappage utilisateur existant pour ajouter, définir ou supprimer l'option de mappage utilisateur `USE_TRUSTED_CONTEXT`. Un utilisateur qui a un mappage utilisateur accrédité peut modifier uniquement l'option `REMOTE_PASSWORD` de son propre mappage utilisateur.

Le scénario

Voici un graphique simple qui illustre un système fédéré mult niveau type qui utilise les mappages utilisateur. Bien que ce scénario inclue un serveur d'application, tout client de base de données peut établir une connexion accréditée.

Emetteur de la connexion : BOSS



Ce scénario comprend quatre utilisateurs :

- Le BOSS qui est l'émetteur de la connexion et a le même ID utilisateur et mot de passe sur le serveur fédéré et sur le serveur de base de données DB2 distant. Par conséquent, le BOSS ne nécessite pas de mappage utilisateur.
- Mary n'a pas de mappage utilisateur. Elle utilise le même ID utilisateur sur le serveur de fédération et sur le serveur de base de données DB2. Dans la mesure où le contexte accrédité spécifie que PUBLIC peut réutiliser la connexion sans authentification, le mot de passe de Mary n'est pas nécessaire.
- Alice a un mappage utilisateur accrédité qui spécifie l'option REMOTE_AUTHID. Alice a un ID utilisateur différent sur le serveur fédéré et sur le serveur de base de données DB2 distant. Dans la mesure où le contexte accrédité spécifie que toute personne peut réutiliser la connexion sans authentification, le mot de passe d'Alice n'est pas nécessaire.
- Sur le serveur fédéré, Emma utilise l'ID utilisateur EMMA. Cet ID utilisateur mappe vers l'ID utilisateur EGREENE et le mot de passe MYPASS sur le serveur de base de données DB2. Dans la mesure où le contexte accrédité spécifie qu'Emma doit s'authentifier, Emma a un mappage utilisateur accrédité qui spécifie à la fois l'option REMOTE_AUTHID et l'option REMOTE_PASSWORD.

Ce scénario comprend trois serveurs :

- Le serveur d'application qui héberge l'application d'assurance et possède l'adresse IP 9.44.111.111.
- Le serveur de fédération, qui a l'adresse IP 9.44.111.222.
- Le serveur de base de données DB2 distant, qui est catalogué sur le serveur fédéré en tant que JUPITER.

Pour configurer ce scénario, le SECADM exécute les étapes suivantes :

1. Sur le serveur de base de données DB2 distant, créez l'objet de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_DB2_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR EMMA WITH AUTHENTICATION,
PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Ce contexte accrédité spécifie que le BOSS est l'émetteur de la connexion accréditée et que la demande de connexion doit provenir du serveur fédéré dont l'adresse IP est 9.44.111.222. Une fois la connexion accréditée établie, tout utilisateur valide de la base de données peut réutiliser la connexion en fournissant uniquement un ID utilisateur.

2. Sur le serveur fédéré, créez cet objet de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_WFS_TCX
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.111')
WITH USE FOR EMMA WITH AUTHENTICATION,
PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Ce contexte accrédité spécifie que le BOSS est l'émetteur de la connexion accréditée et que la demande de connexion doit provenir de l'adresse IP 9.44.111.111, qui identifie le serveur d'application. Une fois la connexion accréditée établie, Emma peut réutiliser ma connexion mais elle doit s'authentifier. Tout utilisateur de base de données valide doit fournir uniquement un ID utilisateur pour réutiliser la connexion.

3. Sur le serveur de fédération, créez cette définition de serveur :

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda
OPTIONS(DBNAME 'remotedb', ...);
```

Cette définition de serveur contient les informations dont le serveur fédéré a besoin pour se connecter à la base de données DB2 distante dénommée remotedb.

4. Sur le serveur de fédération, créez cet objet de contexte accrédité pour Alice :

```
CREATE MAPPING FOR USER ALICE
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'AJACKSON', USE_TRUSTED_CONTEXT 'Y');
```

Ce mappage utilisateur spécifie qu'une connexion accréditée peut être réutilisée par l'utilisateur ALICE qui mappe vers l'ID utilisateur AJACKSON et sur le serveur de base de données DB2 distant.

5. Sur le serveur de fédération, créez ce mappage utilisateur accrédité pour Emma:

```
CREATE MAPPING FOR USER EMMA
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'EGREENE', REMOTE_PASSWORD 'MYPASS', USE_TRUSTED_CONTEXT 'Y');
```

Ce mappage utilisateur spécifie qu'une connexion accréditée peut être réutilisée par l'utilisateur EMMA qui mappe vers l'ID utilisateur EGREENE et sur le serveur de base de données DB2 distant.

Le scénario, étape par étape

1. Le serveur d'application demande une connexion entrante accréditée pour le BOSS.
2. Le BOSS exécute une tâche et l'ID utilisateur du BOSS est propagé via le serveur fédéré vers le serveur de base de données DB2, où les actions effectuées par le BOSS peuvent être auditées.
3. Emma se connecte dans l'application d'assurance qui se trouve sur le serveur d'application. Le serveur d'application demande que la connexion entrante fédérée soit basculée depuis le BOSS vers Emma, après l'authentification d'Emma.
4. Emma exécute une tâche dans l'application.
5. Le serveur fédéré bascule la connexion sortante du BOSS vers Emma, et son ID mappé et mot de passe sont propagés via le serveur fédéré vers le serveur DB2, où les actions effectuées par EGREENE (l'ID utilisateur distant d'Emma) peuvent être auditées.
6. Alice se connecte à l'application d'assurance. Le serveur d'application demande que la connexion entrante fédérée soit basculée d'Emma vers Alice, sans authentifier Alice.
7. Alice exécute une tâche dans l'application.
8. Le serveur fédéré bascule la connexion sortante fédérée d'Emma vers AJACKSON (L'Id utilisateur mappé d'Alice), et son ID est propagé via le serveur fédéré au serveur DB2, où les actions effectuées par AJACKSON peuvent être auditées.
9. Mary se connecte à l'application d'assurance. Mary n'a pas besoin d'authentification ; par conséquent, le serveur d'application bascule la connexion accréditée entrante fédérée vers Mary, sans fournir de mot de passe.
10. Mary exécute une tâche dans l'application.
11. Le serveur fédéré bascule la connexion sortante d'AJACKSON vers Mary, et l'ID utilisateur de Mary est propagé via le serveur fédéré au serveur DB2, où les actions effectuées par Mary peuvent être auditées.

Scénario : Connexions accréditées sortantes fédérées

Une connexion accréditée sortante fédérée établit un environnement accrédité entre le serveur fédéré et un serveur de base de données DB2 distant. Dans la mesure où cette connexion ne nécessite pas d'authentification, vous pouvez éliminer la nécessité de stocker et de conserver les mots de passe.

Pour certaines configurations, le serveur de fédération doit gérer les connexions sortantes qui ne sont pas accréditées. Une connexion n'est pas accréditée lorsque l'une ou l'autre des affirmations ci-dessous est vraie :

- Les attributs de la demande de connexion entrante ne correspondent pas aux attributs de tout objet de contexte accrédité sur le serveur fédéré.
- Le serveur de fédération ne spécifie pas de contexte accrédité pour le serveur d'où provient la demande de connexion. Tous les utilisateurs ont des connexions entrantes non accréditées.

Contexte accrédité, définition du serveur, et exigences de mappage utilisateur

Lorsque vous créez le contexte accrédité sur le serveur de source de données DB2 distant, vous devez spécifier WITH USE FOR PUBLIC WITHOUT AUTHENTICATION. Puis tous les utilisateurs qui utilisent le même ID utilisateur sur le serveur fédéré et sur le serveur de base de données DB2 peuvent utiliser le contexte accrédité sortant sans authentification.

Pour créer la connexion accréditée sortante fédérée, vous spécifiez l'option FED_PROXY_USER dans la définition du serveur pour le serveur de source de données DB2 distant. Cette option identifie l'ID d'autorisation de l'utilisateur qui émet la connexion accréditée sortante. De plus, vous créez un mappage utilisateur pour l'utilisateur de proxy fédéré. Ce mappage utilisateur doit spécifier les deux options REMOTE_AUTHID et REMOTE_PASSWORD car le contexte accrédité sur une source de données nécessite que l'émetteur de la connexion s'authentifie.

Seul un utilisateur bénéficiant des droits d'accès SECADM peut créer et modifier une définition de serveur dont l'option FED_PROXY_USER est définie. De plus, l'instruction SET SERVER OPTION n'est pas valide pour l'option de serveur FED_PROXY_USER.

Il peut y avoir des situations dans lesquelles vous voulez configurer différents ensembles d'utilisateurs pour qu'ils se connectent via différents utilisateurs de proxy. Par exemple, vous pouvez vouloir configurer différents rôles pour différents utilisateurs. Pour permettre cela, dans chaque mappage utilisateur qui n'utilise pas l'utilisateur de proxy fédéré par défaut, vous spécifiez l'option FED_PROXY_USER, et définissez l'option USE_TRUSTED_CONTEXT sur 'Y'. Lorsque l'option FED_PROXY_USER est spécifiée dans la définition du serveur et le mappage utilisateur, la valeur dans le mappage utilisateur remplace la valeur dans la définition du serveur.

Seul un utilisateur bénéficiant des droits d'accès SECADM peut ajouter, supprimer ou définir l'option FED_PROXY_USER et créer ou supprimer un mappage utilisateur comprenant l'option FED_PROXY_USER.

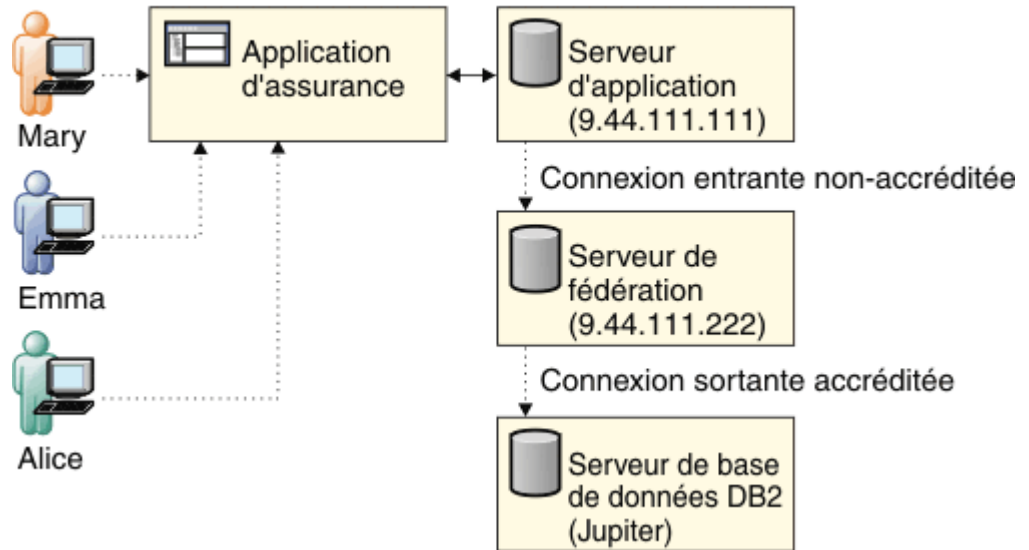
Important :

Il n'y a que deux situations dans lesquelles l'utilisateur d'une connexion sortante accréditée exige un mappage utilisateur : lorsque l'utilisateur utilise un ID utilisateur différent sur le serveur fédéré et sur le serveur de base de données et lorsque l'utilisateur doit se connecter via un proxy fédéré qui n'est pas l'utilisateur de proxy fédéré par défaut.

Le scénario

La figure suivante illustre un scénario qui exige des connexions accréditées sortantes fédérées. Bien que ce scénario inclue un serveur d'application, tout client de base de données peut établir une connexion accréditée.

Utilisateurs de proxy fédéré : BOSS et ADM



Ce scénario comprend cinq utilisateurs :

- le BOSS, qui est l'utilisateur de proxy fédéré par défaut, et possède un mappage utilisateur.
- ADM, qui est un utilisateur de proxy fédéré, et possède un mappage utilisateur.
- Mary qui utilise l'application d'assurance et n'a pas de mappage utilisateur.
- Emma et Alice, dont chacune utilise l'application d'assurance et a un mappage utilisateur.

Ce scénario comprend trois serveurs :

- Le serveur d'application qui héberge l'application d'assurance et possède l'adresse IP 9.44.111.111. Ce scénario présente un serveur d'application, mais tout client peut être utilisé.
- Le serveur de fédération, qui a l'adresse IP 9.44.111.222.
- Le serveur de base de données DB2 distant, qui est catalogué sur le serveur fédéré en tant que JUPITER.

Ces étapes décrivent la configuration :

Remarque : Dans les commandes, les noms d'objet qui sont variables s'affichent en italique. Lorsque vous implémentez des contextes accrédités, spécifiez les noms de variables qui s'appliquent à votre configuration système spécifique.

1. Sur le serveur de base de données DB2 distant, créez ces objets de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_DB2_TXC
BASED UPON CONNECTION USING
SYSTEM AUTHID BOSS
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION
ENABLE
```

Ce contexte accrédité spécifie que le BOSS est l'émetteur de la connexion accréditée et que la demande de connexion accréditée sortante doit provenir du serveur fédéré dont l'adresse IP est 9.44.111.222. Une fois la connexion accréditée établie, tout utilisateur peut réutiliser la connexion en fournissant uniquement un ID utilisateur.

```
CREATE TRUSTED CONTEXT MY_DB2_TXC_ALICE
BASED UPON CONNECTION USING
SYSTEM AUTHID ADM
ATTRIBUTES (ADDRESS '9.44.111.222')
WITH USE FOR PUBLIC WITHOUT AUTHENTICATION ROLE Manager
ENABLE
```

Ce contexte accrédité spécifie que l'ADM est l'émetteur de la connexion accréditée et que la demande de connexion accréditée sortante doit provenir du serveur fédéré dont l'adresse IP est 9.44.111.222. Une fois la connexion accréditée établie, Alice peut réutiliser la connexion sans s'authentifier, et elle obtient le rôle de Manager dans le cadre de la connexion accréditée.

2. Sur le serveur de fédération, créez cette définition de serveur :

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb', FED_PROXY_USER 'BOSS');
```

Cette définition de serveur contient les informations dont le serveur fédéré a besoin pour se connecter à la base de données DB2 distante dénommée remotedb. La définition spécifie que si la connexion entrante au serveur fédéré n'est pas accréditée, alors le BOSS, l'utilisateur du proxy fédéré par défaut, établit la connexion accréditée sortante.

3. Sur le serveur de fédération, créez les mappages utilisateur pour les utilisateurs de proxy fédéré :

```
CREATE USER MAPPING FOR BOSS SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'BOSS', REMOTE_PASSWORD 'MYPASS');
CREATE USER MAPPING FOR ADM SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'ADM', REMOTE_PASSWORD 'PWD');
```

4. Sur le serveur fédéré, créez les mappages utilisateur pour Emma et Alice :

```
CREATE USER MAPPING FOR EMMA SERVER JUPITER
OPTIONS(REMOTE_AUTHID 'EGREENE', USE_TRUSTED_CONTEXT 'Y')
CREATE USER MAPPING FOR ALICE SERVER JUPITER
OPTIONS (USE_TRUSTED_CONTEXT 'Y', FED_PROXY_USER 'ADM');
```

Les deux mappages utilisateur ont l'option USE_TRUSTED_CONTEXT définie sur 'Y', de façon à ce que les utilisateurs puissent utiliser le contexte accrédité. D'autre part, Emma et Alice nécessitent les options suivantes :

- Emma exige un mappage utilisateur car elle n'utilise pas le même ID utilisateur sur le serveur fédéré et sur le serveur de base de données DB2. C'est pourquoi son mappage utilisateur spécifie l'option REMOTE_AUTHID.
- Alice nécessite un mappage utilisateur parce que le contexte accrédité spécifie qu'elle utilise ADM comme utilisateur de proxy fédéré. C'est pourquoi son mappage utilisateur spécifie l'option FED_PROXY_USER.

Dans ce scénario, Mary ne nécessite pas de mappage utilisateur. Les mappages utilisateur pour Emma et Alice ne stockent pas les mots de passe distants ; c'est pourquoi ces mappages utilisateur n'ont pas besoin de mises à jour constantes pour garder les mots de passe à jour.

Le scénario, étape par étape

Ces étapes décrivent comment les connexions sortantes accréditées fonctionnent dans ce scénario.

1. L'application crée une connexion entrante non accréditée vers le serveur fédéré :

```
CONNECT TO FEDSVR USER MARY USING '****'
```
2. La première demande fédérée qui accède au serveur JUPITER crée une connexion sortante vers JUPITER :

```
SELECT * FROM JUPITER_NN01  
CONNECT RESET
```
3. Dans la mesure où JUPITER spécifie FED_PROXY_USER=BOSS, et que le mappage utilisateur de Mary ne spécifie pas d'utilisateur de proxy fédéré, la connexion sortante est créée pour le BOSS puis est immédiatement basculée sur l'utilisateur entrant en cours, qui, dans ce cas, est Mary.
4. La demande fédérée de Mary est complétée et enregistrée dans le journal d'audit sous le nom MARY. Puis la connexion est réinitialisée :
5. Créez une connexion entrante non accréditée au serveur fédéré pour Emma:

```
CONNECT TO FEDSVR USER EMMA USING '****'
```
6. La première demande fédérée dans la connexion en cours qui accède à JUPITER crée une connexion sortante fédérée vers JUPITER.

```
SELECT * FROM JUPITER_NN01  
CONNECT RESET
```
7. La connexion sortante est créée à l'aide du BOSS puis est immédiatement basculée sur l'utilisateur entrant en cours, Emma, dont l'ID est mappé sur EGREENE.
8. La demande fédérée d'Emma est complétée et enregistrée dans le journal d'audit sous le nom EGREENE. Puis la connexion est réinitialisée :
9. Créez une connexion entrante non accréditée au serveur fédéré pour Alice :

```
CONNECT TO FEDSVR USER ALICE USING '****'
```
10. La première demande fédérée dans la connexion en cours qui accède à JUPITER crée une connexion sortante fédérée vers JUPITER. Dans la mesure où le mappage utilisateur accrédité d'Alice spécifie qu'elle utilise l'ADM utilisateur de proxy fédéré plutôt que le BOSS utilisateur de proxy fédéré par défaut, la connexion sortante est créée à l'aide d'ADM puis est immédiatement basculée sur Alice, qui, dans ce cas, est l'utilisateur entrant en cours.

```
SELECT * FROM JUPITER_NN01  
CONNECT RESET
```
11. La demande fédérée d'Alice est complétée et enregistrée dans le journal d'audit. Puis la connexion est réinitialisée :

Mappages utilisateurs et connexions accréditées fédérées

Ces tableaux décrivent comment les mappages utilisateurs, avec et sans ID distants et mots de passe distants, sont utilisés dans des connexions accréditées fédérées de bout en bout et dans les connexions accréditées fédérées sortantes.

Ces tableaux décrivent les mappages utilisateur possible pour BOSS, l'émetteur de la connexion, et pour Mary, le nouvel utilisateur de la connexion. Dans les tableaux, le terme *connexion non accréditée* fait référence à une connexion normale qui n'est pas accréditée sur la connexion entrante et n'est pas accréditée sur la connexion sortante.

Certaines cellules du tableau utilisent le terme *si disponible* pour décrire un mot de passe. Un mot de passe est à la disposition du serveur fédéré si un ID utilisateur et un mot de passe sont explicitement spécifiés dans le cadre de la connexion. Si vous utilisez une API, connectez les appels pour vous connecter au serveur fédéré, puis le mot de passe est explicitement transmis ; par exemple, dans CLI/ODBC, le mot de passe est spécifié via l'attribut l'attribut de connexion SQL_ATTR_TRUSTED_CONTEXT_PASSWORD. Si vous utilisez l'instruction CONNECT pour vous connecter au serveur fédéré, puis utilisez cette syntaxe pour pour transmettre le mot de passe explicitement :

```
CONNECT TO nom de base de données USER ID utilisateur USING mot de passe
```

Mappages pour les émetteurs de connexion et les utilisateurs de proxy fédéré

Tableau 23. Les mappages utilisateur pour BOSS, l'émetteur de la connexion

Mappage utilisateur pour BOSS	Connexion non accréditée	Connexion accréditée fédérée de bout en bout (Lorsque BOSS établit la connexion entrante accréditée)	Connexion accréditée fédérée sortante (Lorsque l'option du serveur ou de mappage utilisateur FED_PROXY_USER='BOSS' est définie)
Pas de mappage utilisateur	Transmettez l'ID utilisateur fédéré et le mot de passe fédéré de BOSS (si disponible) à la source de données distante.	Transmettez l'ID utilisateur fédéré et le mot de passe fédéré de BOSS (si disponible) à la source de données distante.	ERROR SQL1101N
Mappage utilisateur qui spécifie uniquement un ID utilisateur distant	Transmettez l'ID utilisateur distant et le mot de passe fédéré de BOSS (si disponible) à la source de données distante.	Transmettez l'ID utilisateur distant et le mot de passe fédéré de BOSS (si disponible) à la source de données distante.	ERROR SQL1101N
Le mappage utilisateur qui spécifie un ID utilisateur distant et un mot de passe distant	Transmettez l'ID utilisateur distant et le mot de passe distant de BOSS (si disponible) à la source de données distante.	Transmettez l'ID utilisateur distant et le mot de passe distant de BOSS (si disponible) à la source de données distante.	Transmettez l'ID utilisateur distant et le mot de passe distant de BOSS (si disponible) à la source de données distante.

Mappages pour les nouveaux utilisateurs de connexion

Tableau 24. Mappages utilisateur pour Mary, le nouvel utilisateur de la connexion

Mappage utilisateur pour Mary	La connexion accréditée fédérée de bout en bout (Lorsque BOSS établit la connexion accréditée entrante, et que la connexion bascule sur Mary)	Connexion accréditée fédérée sortante (Lorsque l'option du serveur ou de mappage utilisateur FED_PROXY_USER='BOSS' est définie)
Pas de mappage utilisateur	Transmettez l'ID utilisateur fédéré et le mot de passe fédéré de Mary (si disponible) à la source de données distante.	Transmettez l'ID utilisateur fédéré à la source de données distante.
Mappage utilisateur non accrédité qui spécifie uniquement un ID utilisateur distant	Transmettez l'ID utilisateur fédéré et le mot de passe fédéré de Mary (si disponible) à la source de données distante.	Transmettez l'ID utilisateur fédéré de Mary à la source de données distante.
Le mappage utilisateur non accrédité qui spécifie un ID utilisateur distant et un mot de passe distant	Transmettez l'ID utilisateur fédéré et le mot de passe fédéré de Mary (si disponible) à la source de données distante.	Transmettez l'ID utilisateur fédéré de Mary à la source de données distante.
Mappage utilisateur accrédité qui spécifie uniquement un ID utilisateur distant	Transmettez l'ID utilisateur distant de Mary à la source de données distante.	Transmettez l'ID utilisateur distant de Mary à la source de données distante.
Le mappage utilisateur accrédité qui spécifie un ID utilisateur distant et un mot de passe distant	Transmettez l'ID utilisateur distant et le mot de passe distant de Mary à la source de données distante.	Transmettez l'ID utilisateur distant et le mot de passe distant de Mary à la source de données distante.

Chapitre 32. Contrôle d'accès basé sur les intitulés (LBAC) et systèmes fédérés

Veillez à ce que seuls les utilisateurs ayant les droits d'accès appropriés puissent voir les données dans une table.

Grâce au contrôle d'accès basé sur les intitulés, vous pouvez appliquer des politiques de sécurité aux lignes et colonnes d'une table. Chaque politique de sécurité spécifie les droits d'accès qui sont accordés à chaque ID utilisateur et ID de session. Par exemple, la table Prix possède les colonnes Gros, Détail et Soldes. Si l'utilisateur ALice est habilité à accéder aux colonnes Détail et Soldes, alors la requête `SELECT RETAIL, SALE FROM PRICES` réussit. Mais la requête `SELECT * WHOLESALE` échoue.

Lorsque vous créez un pseudonyme dans un objet de source de données Oracle, le serveur fédéré détermine automatiquement si la source de données utilise le contrôle d'accès basé sur les intitulés. Si le contrôle d'accès basé sur les intitulés est utilisé, le pseudonyme n'est pas mis en mémoire cache. Pour les pseudonymes créés avant que le contrôle d'accès basé sur les intitulés ne soit pris en charge, vous devez utiliser l'instruction `ALTER NICKNAME` pour autoriser ou refuser la mise en mémoire cache. Par exemple, si vous avez créé un pseudonyme sur un objet de source de données avant que la prise en charge fédérée du contrôle d'accès basé sur les intitulés ne soit disponible, vous pouvez modifier le pseudonyme pour refuser la mise en mémoire cache.

Chaque politique de sécurité a un intitulé unique qui est stocké dans la colonne Intitulé de la table. Un administrateur de base de données peut masquer la colonne qui contient les intitulés afin d'éviter que les utilisateurs ne sachent que la colonne existe. Les pseudonymes qui ont une colonne d'intitulé masquée ne sont pas mis en mémoire cache.

Chapitre 33. Référentiels de mappage utilisateur externes

Stocke les mappages utilisateurs dans un seul référentiel externe que partagent de nombreux serveurs fédérés, et réduit la maintenance administrative nécessaire lorsque les ID et mots de passe utilisateur sont enregistrés sur chaque serveur fédéré.

Dans une perspective de maintenance, le stockage des mappages utilisateur dans un référentiel externe est plus adéquat que de les stocker dans le catalogue. Les mots de passe distants expirent régulièrement. Si vous stockez les mappages utilisateur dans le catalogue, lorsqu'un mot de passe arrive à expiration, il doit être mis à jour dans la source de données distante et dans le catalogue. Avec un référentiel externe, lorsqu'un mot de passe arrive à expiration, vous ne le mettez à jour qu'une seule fois, dans le référentiel externe.

Pour utiliser un référentiel externe pour les mappages utilisateur, vous devez créer un plug-in de mappage utilisateur. Le plug-in doit utiliser les paramètres de sécurité qu'utilise le référentiel externe. Utilisez Secure Sockets Layer (SSL) pour sécuriser les communications entre le serveur fédéré et le référentiel externe. Puis, lorsque vous créez le fichier de configuration pour le plug-in de mappage utilisateur, spécifiez que le plug-in utilise SSL. De même, limitez l'accès au code source du plug-in de sorte que les informations restent accréditées.

Si vous activez la fonction d'audit, chaque fois que le serveur fédéré utilise le plug-in de mappage utilisateur, un enregistrement d'audit VALIDATE est créé. Pour capturer les enregistrements VALIDATE, configurez la fonction db2audit.

Ces plug-ins échantillons et instructions complètes pour la création, le test, et le déploiement de votre propre plug-in personnalisé dans fournis :

Plug-in de mappage utilisateur (langage de programmation C)

Le plug-in se compose de cinq fonctions qui fournissent une interface afin d'extraire les mappages utilisateurs d'un référentiel externe.

La séquence de l'appel de fonction est la suivante

1. FSUMPluginInit
2. FSUMconnect
3. FSUMfetchUM
4. FSUMdisconnect
5. FSUMPluginTerm

Initialisation – FSUMPluginInit

Dès que le serveur fédéré charge la bibliothèque de plug-ins, le serveur appelle la fonction FSUMPluginInit, qui initialise le plug-in et transmet au serveur fédéré les pointeurs pour les autres fonctions. Ces fonctions sont globales et doivent être résolues en externe. Si le plug-in est écrit en C++, ces fonctions doivent être déclarées avec un "C" externe. Le serveur fédéré transmet les pointeurs à un ensemble de fonctions de programme utilitaire, que le plug-in peut obtenir selon les besoins.

Le plug-in est chargé dans un processus d'unités d'exécution db2fmp. Toutes les applications qui utilisent le plug-in partagent la même bibliothèque, qui doit autoriser les unités d'exécution multiples. Chaque application utilise une unité d'exécution dans le processus db2fmp. Dès que l'application parvient à extraire un mappage utilisateur et à le nettoyer, le serveur fédéré renvoie le pool d'unités d'exécution pour une utilisation ultérieure. Dans la mesure où le serveur fédéré peut servir simultanément plusieurs applications, plusieurs unités d'exécution qui partagent la même bibliothèque de plug-ins peuvent être activées en même temps. Chaque unité d'exécution possède un descripteur de connexion distinct au référentiel de mappage utilisateur. L'API FSUMPluginInit fournit également un moyen pour que chaque unité d'exécution gère des ressources de plug-in globales, par exemple, pour augmenter le décompte de référence au plug-in.

Connexion au référentiel – FSUMconnect

Le serveur fédéré appelle la fonction FSUMconnect pour se connecter au référentiel de mappage utilisateur. Le plug-in doit inclure un descripteur qui enregistre toutes les informations nécessaires pour établir la connexion au référentiel. Ainsi, les informations peuvent inclure un fichier ouvert ou un descripteur de connexion à un serveur LDAP. Selon la manière dont vous implémentez la sécurité du référentiel externe, les informations peuvent inclure également un ID utilisateur et un mot de passe. Si des droits d'accès sont nécessaires, vous pouvez établir la manière dont ceux-ci sont gérés. Par exemple, si vous mettez des droits d'accès sur un fichier de configuration, lorsque le plug-in tente de se connecter au référentiel externe, le plug-in lit les droits d'accès à partir de ce fichier.

Extraction du mappage utilisateur – FSUMfetchUM

Le plug-in appelle la fonction FSUMfetchUM pour extraire le mappage utilisateur du référentiel externe et appelle la fonction de programme utilitaire FSUMaddUMOption pour envoyer l'ID distant et le mot de passe distant au serveur fédéré. Dans le référentiel, chaque mappage utilisateur est identifié par le nom d'instance de serveur fédéré, le nom de serveur distant, et l'ID d'autorisation locale. De plus, chaque mappage utilisateur doit inclure les options REMOTE_AUTHID et REMOTE_PASSWORD. Si un mot de passe distant est chiffré, le plug-in doit le déchiffrer avant de l'envoyer au serveur fédéré.

Déconnexion du référentiel – FSUMdisconnect

Le serveur fédéré appelle la fonction FSUMdisconnect pour se déconnecter du référentiel de mappage utilisateur. Pour se déconnecter, cette fonction désassocie l'unité d'exécution du référentiel de mappage utilisateur. Par exemple, cette fonction peut fermer un fichier ouvert ou fermer une connexion à un serveur LDAP.

Libération de ressources globales – FSUMPluginTerm

En dernière étape, le serveur fédéré appelle la fonction FSUMPluginTerm pour libérer toute ressource globale que la fonction FSUMPluginInit a allouée. Pour terminer, cette fonction désassocie l'unité d'exécution du plug-in.

Plates-formes prises en charge pour le plug-in de mappage utilisateur (langage de programmation C)

Avant de générer le plug-in, confirmez que vous utilisez une plate-forme prise en charge.

La table suivante dresse la liste des plates-formes prises en charge pour le plug-in de mappage utilisateur.

Plateforme	Nom du fichier
AIX, 64 bit	plugin_file_name.a
Linux AMD 64, Power PC, 64, Power PC 390	plugin_file_name.so
Microsoft Windows, 32-bit et 64-bit	plugin_file_name.dll

Restrictions au développement d'un plug-in de mappage utilisateur (langage de programmation C)

Lorsque vous développez des plug-ins de mappage utilisateur en langage C, gardez ces restrictions à l'esprit.

Liaison C

La bibliothèque de plug-ins doit être liée avec une liaison C. Les fichiers d'en-tête qui fournissent les prototypes, les structures de données pour implémenter le plug-in et les définitions des codes d'erreur sont fournis uniquement en C/C++. Les fonctions qui seront résolues en temps de chargement doivent être déclarées avec un "C" externe si la bibliothèque de plug-ins est compilée en tant que C++.

la routine d'exécution du langage commun .NET n'est pas prise en charge

La routine d'exécution du langage commun (CLR) .NET n'est pas prise en charge pour la compilation et la liaison du code source pour la bibliothèque de plug-ins.

Gestionnaires de signaux

La bibliothèque de plug-ins ne doit pas installer les gestionnaires de signaux ni modifier le masque de signaux car le faire interfère avec le signalement et la récupération à partir d'erreurs. La bibliothèque de plug-ins ne doit jamais lever des exceptions C++.

Unités d'exécution multiples

La bibliothèque de plug-ins doit autoriser les unités d'exécution multiples et être réentrante. Seule l'initialisation du plug-in n'a pas de nécessité à être réentrante. La fonction d'initialisation du plug-in peut potentiellement être appelée plusieurs fois à partir de différentes unités d'exécution, auquel cas, le plug-in nettoie toutes les ressources utilisées et se réinitialise lui-même.

Substitution de la bibliothèque C standard et des appels du système d'exploitation

La bibliothèque de plug-ins ne doit pas remplacer la bibliothèque C standard et les appels du systèmes d'exploitation.

Applications 32-bit et 64-bit

Un serveur fédéré de 32 bits doit utiliser un plug-in de 32 bits. Un serveur fédéré de 64 bits doit utiliser un plug-in de 64 bits. Dans une instance hybride, lorsque le client est de 32-bit et le serveur de 64-bit, le plug-in doit être de 64-bit.

Chaînes de texte

les chaînes d'entrée ne sont pas garanties être null-terminated, et les chaînes de sortie n'ont pas besoin d'être null-terminated. A la place, les longueur de nombres entiers sont données pour toutes les chaînes d'entrée, et les pointeurs vers des nombres entiers sont données pour que les longueurs soient renvoyées.

Fichier d'en-tête (fsumplugin.h) pour le plug-in de mappage utilisateur (langage de programmation C)

Le fichier d'en-tête, fsumplugin.h, contient des structures de données, fonctions et un code d'erreur.

Le plug-in de mappage utilisateur doit inclure ce fichier d'en-tête, qui se trouve dans le répertoire sqllib/include.

```
/* Définition des noms d'option de mappage utilisateur. */
#définir FSUM_REMOTE_AUTHID_OPTION "REMOTE_AUTHID"
#définir FSUM_REMOTE_PASSWORD_OPTION "REMOTE_PASSWORD"

/* Définition des types de valeur d'option. */
#définir FSUM_OPTION_VALUE_BINARY_TYPE 1
#définir FSUM_OPTION_VALUE_STRING_TYPE 2

/* Structure de données pour décrire une option utilisateur. */

typedef struct _FSUMOption
{
    const char* optionName;
    size_t optionNameLen;
    char* optionValue;
    size_t optionValueLen;
    size_t optionValueType;
    struct _FSUMOption* nextOption;
} FSUMOption;

/* Structure de données pour décrire une entrée de mappage utilisateur.
Une entrée de mappage utilisateur peut avoir plusieurs options de mappage
utilisateur, telles que REMOTE_AUTHID et REMOTE_PASSWORD. Ces options sont
placées dans une liste liée. Cette structure de données contient le
pointeur vers la première option de la liste. */

typedef struct _FSUMEntry
{
    const char* fsInstanceName;
    size_t fsInstanceNameLen;
    const char* fsDatabaseName;
    size_t fsDatabaseNameLen;
    const char* fsServerName;
    size_t fsServerNameLen;
    const char* fsAuthID;
    size_t fsAuthIDLen;
    FSUMOption* firstOption;
} FSUMEntry;

/* Les fonctions à implémenter en plus de la fonction FSUMPluginInit,
qui n'est pas dans la structure FSUMPluginAPIs. */

typedef struct _FSUMPluginAPIs
{
    size_t version;
    SQL_API_RC (SQL_API_FN * FSUMconnect)
    (void** a_FSUMRepository, const char* a_cfgFilePath);

    SQL_API_RC (SQL_API_FN * FSUMfetchUM)
    (void* a_FSUMRepository, FSUMEntry* a_entry);

    SQL_API_RC (SQL_API_FN * FSUMdisconnect)
    (void* a_FSUMRepository);

    SQL_API_RC (SQL_API_FN * FSUMPluginTerm) ();
} FSUMPluginAPIs;
```

```

/* Le serveur fédéré fournit ces utilitaires comme fonctions appelées
   au plug-in. */

typedef SQL_API_RC (SQL_API_FN FSUMallocateFP)
(size_t a_blkSize, void** a_blkPtr);

typedef void(SQL_API_FN FSUMdeallocateFP)
(void* a_blkPtr);

typedef SQL_API_RC (SQL_API_FN FSUMloadFP)
(const char* a_libName, void** a_lib);

typedef SQL_API_RC (SQL_API_FN FSUMgetFunctionFP)
(const char* a_functionName, void* a_lib, void** a_pFuncAddress);

typedef SQL_API_RC (SQL_API_FN FSUMunloadFP)
(void* a_lib);

typedef SQL_API_RC (SQL_API_FN FSUMlogErrorMsgFP)
(sqlint32 a_level, const char* a_msg, size_t a_length);

typedef SQL_API_RC (SQL_API_FN FSUMaddUMOptionFP)
(FSUMEntry *a_entry,
const char* optionName,
size_t optionNameLen,
const char* optionValue,
size_t optionValueLen);

/* Structure pour conserver les fonctions de programme utilitaire
   que fournit le serveur fédéré. */

typedef struct _FSUMPluginUtilities
{
    FSUMallocateFP    *allocate;
    FSUMdeallocateFP *deallocate;
    FSUMloadFP        *load;
    FSUMgetFunctionFP *getFunction;
    FSUMunloadFP      *unload;
    FSUMlogErrorMsgFP *logErrorMsg;
    FSUMaddUMOptionFP *addUMOption;
} FSUMPluginUtilities;

/* Type de point d'entrée du plug-in de mappage utilisateur. */
typedef SQL_API_RC (SQL_API_FN *FSUMPluginInitType)
(sqlint32, FSUMPluginAPIs*, FSUMPluginUtilities*);

/* Type de point d'entrée de l'interface C du plug-in de mappage utilisateur. */
typedef SQL_API_RC (*fsum_plugin_hook_type)
(const char*, FSUMPluginInitType*, FSUMPluginUtilities*);

/* Définition des codes retour pour les fonctions de programme utilitaire */
#définir FSUM_PLUGIN_UTIL_OK 0
#définir FSUM_PLUGIN_UTIL_FAILED -1

/* Définition du C externe. */
#définir FSUM_PLUGIN_EXT_C extern "C"

/* Gravité des erreurs que la fonction logErrorMsg utilise.*/
#définir FSUM_LOG_NONE 0 /* No logging */
#définir FSUM_LOG_CRITICAL 1 /* Severe error encountered */
#définir FSUM_LOG_ERROR 2 /* Error encountered */
#définir FSUM_LOG_WARNING 3 /* Warning */
#définir FSUM_LOG_INFO 4 /* Informational */

/* Codes d'erreurs renvoyés par les fonctions.*/
#définir FSUM_PLUGIN_OK 0

```

```

#définir FSUM_INITIALIZE_ERROR      1
#définir FSUM_PLUGIN_VERSION_ERROR  2
#définir FSUM_CONNECTION_ERROR     3
#définir FSUM_LOOKUP_ERROR         4
#définir FSUM_DECRYPTION_ERROR     5
#définir FSUM_DISCONNECT_ERROR     6
#définir FSUM_INVALID_PARAMETER_ERROR 7
#définir FSUM_UNAUTHORIZED_CALLER  8
#définir FSUM_AUTHENTICATION_ERROR  9
#définir FSUM_TERMINATION_ERROR    10

/* Longueur maximale d'un nom.*/
#définir FSUM_MAX_NAME_LEN          128

/* Longueur maximale d'un chemin d'accès à un fichier. */
#définir FSUM_MAX_PATH_LEN          256

/* Longueur maximale d'une valeur d'option. */
#définir FSUM_MAX_OPTION_VALUE_LEN  (2048+1)

/* Longueur maximale d'un message d'erreur. */
#définir FSUM_MAX_ERROR_MSG_SIZE    2048

```

Fonction FSUMPluginInit (langage de programmation C)

FSUMPluginInit est la fonction d'initialisation du plug-in de mappage utilisateur.

Cette fonction exécute les tâches suivantes :

- Transmet les pointeurs pour les quatre autres fonctions requises au serveur fédéré.
- Obtient les fonctions de programme utilitaire que transmet le serveur fédéré.
- Définit les ressources globales au niveau du plug-in.

Syntaxe

```

SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
FSUMPluginAPIs* a_pluginAPIs,
FSUMPluginUtilities* a_pluginUtils);

```

Entrées

sqlint32 a_version

Le numéro de version de l'interface du plug-in de mappage utilisateur.

FSUMPluginUtilities *a_pluginUtils

Le pointeur vers la structure qui contient toutes les fonctions du programme utilitaire. Le plug-in obtient un pointeur vers chaque fonction du programme utilitaire, selon les besoins.

Sorties

FSUMPluginAPIs *a_pluginAPIs

Le plug-in utilise cette structure pour transmettre les pointeurs de fonctions pour FSUMconnect, FSUMfetchUM, FSUMdisconnect, et FSUMPluginTerm au serveur fédéré.

Fonction FSUMconnect (langage de programmation C)

Pour se connecter au référentiel de mappage utilisateur externe, le serveur fédéré appelle la fonction FSUMconnect.

Syntaxe

```
SQL_API_RC SQL_API_FN FSUMconnect  
(void** a_FSUMRepository,  
const char* a_cfgFilePath)
```

Entrées

const char* a_cfgFilePath

Le chemin d'accès complet à la bibliothèque de plug-in. Si le fichier de configuration est placé dans le même répertoire que la bibliothèque de plug-in, le plug-in peut utiliser ces informations de chemin d'accès pour localiser le fichier de configuration.

Sorties

void a_FSUMRepository**

Le pointeur vers le descripteur de connexion est transtypé sur vide avant d'être envoyé au serveur fédéré. Dans les futurs appels de fonction API, le serveur fédéré transmet le pointeur au plug-in, et le plug-in doit transtyper le pointeur dans la structure réelle du descripteur de connexion.

Fonction FSUMfetchUM (langage de programmation C)

Le serveur fédéré appelle la fonction FSUMfetchUM pour extraire les options de mappage utilisateur à partir d'un référentiel externe.

Chaque entrée de mappage utilisateur est identifiée par le nom d'instance fédérée (fsInstanceName), le nom de base de données (fsDatabaseName), le nom de serveur distant (fsServerName), et l'ID d'autorisation de l'utilisateur local (fsAuthID). Le mappage utilisateur peut également inclure les options REMOTE_AUTHID et REMOTE_PASSWORD, que le plug-in récupère.

Syntaxe

```
SQL_API_RC SQL_API_FN FSUMfetchUM (void* a_FSUMRepository,  
FSUMEntry* a_entry);
```

Entrées

void* a_FSUMRepository

Le descripteur de connexion au référentiel externe. Ce descripteur doit être transtypé à la structure réelle qui est définie dans le plug-in.

FSUMEntry* a_entry

Ce paramètre est à la fois un paramètre d'entrée et un paramètre de sortie.

En tant qu'entrée, ce paramètre transmet les informations qui sont nécessaires pour identifier l'entrée de mappage utilisateur dans le référentiel de mappage utilisateur. Ces informations comprennent fsInstanceName, fsDatabaseName, fsServerName, et fsAuthID.

En tant que sortie, les options de mappage utilisateur qui sont extraites du référentiel de mappage utilisateur sont ajoutées à ce paramètre. Le plug-in doit utiliser la fonction de programme utilitaire FSUMaddUMOption pour ajouter une option à l'entrée de mappage utilisateur, a_entry. Si le mappage utilisateur comprend l'option REMOTE_PASSWORD et si le mot de passe est codé, le plug-in doit déchiffrer le mot de passe avant d'appeler FSUMaddUMOption. Le serveur fédéré n'est pas responsable de la libération de mémoire pour les chaînes qui sont transmises à la fonction FSUMaddUMOption.

Fonction FSUMdisconnect (langage de programmation C)

Pour se déconnecter du référentiel de mappage utilisateur externe, le serveur fédéré appelle cette fonction.

Syntaxe

```
SQL_API_RC SQL_API_FN FSUMdisconnect  
(void* a_FSUMRepository);
```

Entrées

void* a_FSUMRepository

Le descripteur de connexion au référentiel externe. Ce descripteur doit être transtypé à la structure réelle qui est définie dans le plug-in.

Fonction FSUMPluginTerm (langage de programmation C)

Le plug-in de mappage utilisateur appelle cette fonction pour libérer les ressources globales allouées par la fonction FSUMPluginInit.

Cette fonction n'a pas de paramètres.

```
SQL_API_RC SQL_API_FN FSUMPluginTerm ()
```

Développement d'un plug-in de mappage utilisateur (langage de programmation C)

Le plug-in que vous développez doit se connecter au référentiel externe, récupérer les mappages utilisateur, et décrypter les mots de passe distants. Le référentiel que vous utilisez détermine la manière dont vous codez le plug-in.

Important : Sachez qu'à mesure que vous développez et utilisez le plug-in, vous envoyez des mots de passe et ID utilisateur sensibles entre plusieurs sources. Pour protéger ces informations, restreignez l'accès au code source du plug-in, et configurez l'utilitaire de contrôle de DB2 pour capturer un enregistrement VALIDATE dans le fichier journal de diagnostic chaque fois que le serveur fédéré utilise le plug-in. Le fichier journal de diagnostic est utile non seulement pour tracer l'utilisation mais aussi pour le diagnostic de panne de tout incident qui se produit.

Pour développer un plug-in de mappage utilisateur, procédez comme suit :

Déclaration de fonctions externes pour le plug-in de mappage utilisateur (langage de programmation C)

Le plug-in implémente les fonctions. Lorsque la fonction FSUMPluginInit est appelée, elle transmet les pointeurs de la fonction au serveur fédéré.

La fonction d'initialisation, FSUMPluginInit, est la seule fonction qui doit avoir exactement ce nom de fonction. Pour chacune des autres fonctions, vous pouvez utiliser tout nom de fonction C valide. Les fonctions sont globales et doivent être résolues en externe. Si vous écrivez le plug-in en C++, vous devez utiliser FSUM_PLUGIN_EXT_C pour déclarer les fonctions.

Le plug-in échantillon implémente les API suivantes :

- API FSUMconnect dans la fonction myConnect
- FSUMfetchUM dans la fonctions myFetchUM
- FSUMdisconnect dans la fonction myDisconnect
- FSUMPluginTerm dans la fonction myPluginTerm

Dans l'échantillon, lorsque la fonction FSUMPluginInit est appelée, les pointeurs de fonction pour myConnect, myFetchUM, myDisconnect, et myPluginTerm sont transmis au serveur fédéré.

Le code suivant est dans le fichier fsumplugin_file.c.

```
SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint23 version,
FSUMPluginAPIs *pluginAPIs,
FSUMPluginUtilities* pluginUtils)
```

```
SQL_API_RC SQL_API_FN myConnect
(void** FSUMRepository)
```

```
SQL_API_RC SQL_API_FN myFetchUM
(void* FSUMRepository,
FSUMEntry* entry)
```

```
SQL_API_RC SQL_API_FN myDisconnect
(void* FSUMRepository)
```

```
SQL_API_RC SQL_API_FN myPluginTerm ()
```

Déclaration de fonctions de programme utilitaire pour le plug-in de mappage utilisateur (langage de programmation C)

Vous devez déclarer quatre options d'utilitaire requises pour le plug-in de mappage utilisateur.

FSUMlogErrorMsg, FSUMaddUMOption, FSUMallocate, et FSUMdeallocate sont des fonction de programme utilitaire pour le plug-in de mappage utilisateur. Les fonctions de programme utilitaire facultatives peuvent être utiles lorsque vous développez un plug-in de mappage utilisateur. Vous obtenez les pointeurs des fonctions du programme utilitaire auprès de la structure FSUMPluginUtilities dans la fonction FSUMPluginInit.

Fonctions de programme utilitaire requises

```
FSUMlogErrorMsgFP *FSUMlogErrorMsg=NULL;
FSUMaddUMOptionFP *FSUMaddUMOption=NULL;
FSUMallocateFP *FSUMallocate=NULL;
FSUMdeallocateFP *FSUMdeallocate=NULL;
```

Fonctions de programme utilitaire facultatives

```
FSUMgetFunctionFP *FSUMgetFunction=NULL;
FSUMloadFP *FSUMload=NULL;
FSUMunloadFP *FSUMunload=NULL;
```

Définition de pointeurs de fonction pour le plug-in de mappage utilisateur (langage de programmation C)

Transmettez les pointeurs aux fonctions requises sur le serveur fédéré, et obtenez les pointeurs vers les fonctions de programme utilitaire à partir du serveur fédéré.

```
/* La fonction d'initialisation du plug-in. Ne modifiez pas le nom. */
```

```
SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
FSUMPluginAPIs* a_pluginAPIs,
FSUMPluginUtilities* a_pluginUtils)
{
```

```
    SQL_API_RC rc = FSUM_PLUGIN_OK ;
```

```
    /* Pass the fonction pointers to federated server */
```

```
    a_pluginAPIs->FSUMconnect = &myConnect
```

```

a_pluginAPIs->FSUMfetchUM = &myFetchUM
a_pluginAPIs->FSUMdisconnect = &myDisconnect
a_pluginAPIs->FSUMPluginTerm = &myPluginTerm

/* Obtenez les pointeurs des fonctions de programme utilitaire requises */

FSUMallocate = a_pluginUtils->allocate;
FSUMdeallocate = a_pluginUtils->deallocate;
FSUMlogErrorMsg = a_pluginUtils->logErrorMsg;
FSUMaddUMOption = a_pluginUtils->addUMOption;
/*Vous pouvez également obtenir les pointeurs des fonctions de programme
utilitaire facultatives */

/** En cas d'erreur, procédez comme suit :
 1. Facultatif --- Appelez FSUMlogErrorMsg pour journaliser l'erreur
 2. Obligatoire --- Renvoie le code d'erreur rc = FSUM_INITIALIZE_ERROR
**/

renvoi rc;
}

```

Traitement des erreurs dans le plug-in de mappage utilisateur (langage de programmation C)

Le plug-in de mappage utilisateur utilise la fonction FSUMlogErrorMsg pour journaliser tous les messages d'erreur et d'information dans le fichier db2diag.log.

En cas de succès d'un appel de fonction, le plug-in renvoie FSUM_PLUGIN_OK. En cas d'échec de l'appel, le plug-in renvoie un code d'erreur. Le tableau suivant décrit les erreurs.

Tableau 25. Erreurs pour le plug-in de mappage utilisateur

Numéro d'erreur	Nom de la constante	Message d'erreur	Fonctions qui renvoient l'erreur	Code d'erreur
0	FSUM_PLUGIN_OK	Le plug-in s'est exécuté avec succès.	<ul style="list-style-type: none"> FSUMPluginInit FSUMconnect FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL_SUCCESS (Pas d'erreur)
1	FSUM_INITIALIZE_ERROR	Echec de l'initialisation du plug-in.	<ul style="list-style-type: none"> FUMPluginInit 	SQL20349N , code anomalie 1
2	FSUM_PLUGIN_VERSION_ERROR	La version du plug-in est erronée.	<ul style="list-style-type: none"> FUMPluginInit 	SQL20349N, code anomalie 2
3	FSUM_CONNECTION_ERROR	Connexion impossible au référentiel externe.	<ul style="list-style-type: none"> FSUMPluginInit 	SQL20349N, code anomalie 3
4	FSUM_LOOKUP_ERROR	Echec de la recherche sur le référentiel.	<ul style="list-style-type: none"> FSUMfetchUM 	SQL20349N , code anomalie 4
5	FSUM_DECRYPTION_ERROR	Echec du déchiffrement.	<ul style="list-style-type: none"> FSUMfetchUM 	SQL20349N, code anomalie 5
6	FSUM_DISCONNECT_ERROR	Déconnexion du référentiel impossible.	<ul style="list-style-type: none"> FSUMdisconnect 	SQL20349N , code anomalie 6
7	FSUM_INVALID_PARAMETER_ERROR	Paramètre incorrect.	<ul style="list-style-type: none"> FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL20349N, code anomalie 7

Tableau 25. Erreurs pour le plug-in de mappage utilisateur (suite)

Numéro d'erreur	Nom de la constante	Message d'erreur	Fonctions qui renvoient l'erreur	Code d'erreur
8	FSUM_UNAUTHORIZED_CALLER	L'appelant n'est pas autorisé à appeler le plug-in.	<ul style="list-style-type: none"> FSUMPluginInit 	SQL20349N , code anomalie 8
9	FSUM_AUTHENTICATION_ERROR	Authentification avec le référentiel impossible.	<ul style="list-style-type: none"> FSUMconnect 	SQL20350N, pas de code anomalie
10	FSUM_TERMINATION_ERROR	Echec du nettoyage d'une ressource au niveau du plug-in.	<ul style="list-style-type: none"> FSUMPluginTerm 	SQL20349N, code anomalie 9
Pas un numéro de 0 à 10		Une erreur inconnue s'est produite.	<ul style="list-style-type: none"> FSUMPluginInit FSUMconnect FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL20349N, code anomalie 10

Le plug-in peut également utiliser la fonction de programme utilitaire FSUMlogErrorMsg pour journaliser les messages dans le fichier db2diag.log. Pour spécifier la gravité des messages, utilisez FSUM_LOG_CRITICAL, FSUM_LOG_ERROR, FSUM_LOG_WARNING, ou FSUM_LOG_INFO.

Lorsque vous utilisez le programme de test pour tester le plug-in, les messages sont imprimés directement sur l'écran.

Génération du plug-in de mappage utilisateur (C)

Utilisez le script bldplugin pour générer le plug-in de mappage utilisateur.

Pour générer le plug-in de mappage utilisateur :

1. Examinez le script bldplugin et mettez à jour les informations du chemin d'accès afin de correspondre à votre installation spécifique.
2. Emettez cette commande pour générer le plug-in :

```
bldplugin nom_fichier_plugin
```

Pour plus d'informations, voir le fichier bldplugin.
3. Configurez le référentiel externe de manière à inclure les informations de mappage utilisateur et d'appliquer un schéma de chiffrement. Si vous générez le plug-in de mappage utilisateur échantillon, ouvrez le fichier fsumplugin_file.txt et créez une entrée de mappage pour votre système. Le plug-in échantillon utilise un schéma de chiffrement simple qui inverse les octets du mot de passe distant.

Test du plug-in de mappage utilisateur (langage de programmation C)

Utilisez le programme de test pour charger et tester le plug-in d'exemple de mappage utilisateur ou votre propre plug-in de mappage utilisateur personnalisé afin de la déployer sur le serveur fédéré.

Le programme d'installation fsumsetup_file (fsumsetup_file.exe sur Microsoft Windows) et le programme de test fsumlookup (fsumlookup.exe sur Windows) sont installés dans le même répertoire où sont installés les fichiers du plug-in d'exemple.

Le programme `fsumsetup_file` génère un fichier de configuration dénommé `fsumplugin_file.cfg`. Ce fichier stocke le chemin d'accès complet au fichier texte qui contient les entrées du mappage utilisateur. Le programme `fsumlookup` (`fsumlookup.exe` sur Microsoft Windows) teste le plug-in.

1. Exécutez le programme `fsumsetup_file` pour configurer le chemin d'accès complet et le nom du fichier qui stocke les mappages utilisateur.
2. Exécutez le programme `fsumlookup` pour tester le plug-in :

```
fsumlookup pluginName fsInstance fsDatabase fsRemoteServer fsAuthid
```

où :

- *pluginName* est le nom du plug-in.
- *fsInstance* est le nom de l'instance de serveur fédéré.
- *fsDatabase* est le nom de la base de données fédérée.
- *fsRemoteServer* est le nom du serveur de source de données distant, tel que spécifié dans l'instruction CREATE SERVER.
- *fsAuthid* est l'ID utilisateur local.

Si le plug-in fonctionne correctement, le programme `fsumlookup` imprime les paramètres qui identifient l'entrée de mappage utilisateur, ainsi que les noms d'option et les valeurs d'option, à l'écran. Ce qui suit est un exemple de la sortie.

```
fsumlookup fsumplugin_file.a FSinst1 FSdb remoteDB localID
fsInstanceName: FSinst1
fsDatabaseName: FSdb
fsServerName: remoteDB
fsAuthID: localID
optionName:REMOTE_PASSWORD
optionValue:p4ssw0rd
optionName:REMOTE_AUTHID
optionValue:remoteID
```

Si un message d'erreur est généré, le programme de test imprime le message d'erreur à l'écran.

Déploiement du plug-in de mappage utilisateur (langage de programmation C)

Vous pouvez générer et compiler un plug-in de mappage utilisateur dans tout répertoire. Puis, pour déployer le plug-in, copiez-le dans le répertoire adéquat sur le serveur fédéré.

Pour déployer le plug-in sur le serveur fédéré, copiez le plug-in et le fichier de configuration sur le répertoire adéquat sur le serveur fédéré :

- Sous UNIX et Linux, `inst_home/sql/lib/function`, où `inst_home` est le répertoire initial d'instance
- Sous Windows, `%DB2PATH%\function`, où `%DB2PATH%` est le répertoire où est installé le système de base de données DB2

Activer l'accès au référentiel de mappage utilisateur externe

Après avoir créé le plug-in de mappage utilisateur, vous devez définir l'option `DB2_UM_PLUGIN` et `DB2_UM_PLUGIN_LANG` pour activer le serveur fédéré pour accéder aux mappages utilisateur stockés dans le référentiel externe.

Vous pouvez définir les options au niveau de l'encapsuleur ou du serveur, mais vous devez définir les deux options au même niveau. `DB2_UM_PLUGIN` spécifie le nom du plug-in et `DB2_UM_PLUGIN_LANG` spécifie le langage dans lequel est écrit le plug-in. Les deux options dépendent l'une de l'autre. C'est pourquoi vous devez ajouter `DB2_UM_PLUGIN` avant d'ajouter `DB2_UM_PLUGIN_LANG`. Si

vous ajoutez les deux options dans la même instructions, l'ordre dans lequel vous les spécifiez n'a pas d'importance. Pour supprimer les options, vous devez supprimer DB2_UM_PLUGIN_LANG avant de supprimer DB2_UM_PLUGIN.

Par exemple, les instructions suivantes utilisent le nom du plug-in échantillon pour créer l'encapsuleur w1, modifier l'encapsuleur w2, créer le serveur s1 et modifier le serveur s2 pour utiliser le plug-in de mappage utilisateur :

```
CREATE WRAPPER w1 LIBRARY 'libdb2drda.a'
  OPTIONS
  (DB2_UM_PLUGIN 'fsumplugin_file.a', DB2_UM_PLUGIN_LANG 'C');

ALTER WRAPPER w2
  OPTIONS
  (ADD DB2_UM_PLUGIN 'fsumplugin_file.a', ADD DB2_UM_PLUGIN_LANG 'C');

CREATE SERVER s1 TYPE db2/cs VERSION 10 WRAPPER w2
  AUTHORIZATION remoteID PASSWORD p4ssw0rd
  OPTIONS
  (NODE 'node1',
   DBNAME 'db1',
   DB2_UM_PLUGIN 'fsumplugin_file.a',
   DB2_UM_PLUGIN_LANG 'C');

ALTER SERVER s2
  OPTIONS
  (ADD DB2_UM_PLUGIN 'fsumplugin_file.a', ADD DB2_UM_PLUGIN_LANG 'C');
```

Mise à jour du plug-in de mappage utilisateur (langage de programmation C)

Après avoir modifié un plug-in de mappage utilisateur existant, vous devez copier la version mise à jour dans le serveur fédéré.

Pour mettre à jour le plug-in de mappage utilisateur :

1. Copiez le plug-in d'exemple dans le répertoire sqllib/function.
2. Entrez ces commandes pour arrêter puis redémarrer l'instance de serveur fédéré :

```
db2stop
db2start
```

Echantillon de plug-in de mappage utilisateur (en langage de programmation C)

Révisez le code source exemple pour apprendre comment développer un plug-in qui fournit une interface C à un référentiel de mappages utilisateur externe.

Ce plug-in exemple de mappage utilisateur utilise un fichier en tant que référentiel externe. Les fichiers source exemples sont automatiquement installés sur votre système. L'emplacement où ils sont installés dépend du système d'exploitation que vous utilisez :

- Sous Windows, %DB2PATH%\samples\federated\umplugin\file, où %DB2PATH% est le répertoire où est installé le système de base de données DB2
- Sous Linux et UNIX, *inst_home*/sqllib/samples/federated/umplugin/file, où *inst_home* est le répertoire initial d'instance

L'exemple comprend les fichiers suivants :

fsumplugin_file.h

Le fichier d'en-tête qui contient les structures de données et les définitions de fonction pour le plug-in d'exemple.

fsumplugin_file.c

Le code source du plug-in d'exemple, qui utilise un fichier en tant que référentiel externe.

fsumplugin_file.txt

Un fichier qui contient les entrées du mappage utilisateur d'exemple qui sont en texte en clair.

fsumlookup

Un programme autonome qui teste le plug-in d'exemple en dehors de l'environnement de serveur fédéré. Sous Microsoft Windows, le fichier est fsumlookup.exe.

fsumsetup_file

Un programme autonome qui définit le chemin d'accès complet et le nom de fichier qui stocke les mappages utilisateur. Sous Microsoft Windows, le fichier est fsumsetup_file.exe

bldplugin

Le script de génération qui compile et lie le plug-in. Sous Microsoft Windows, le fichier est bldplugin.bat.

README

Un fichier qui contient des instructions pour compiler, générer, tester et déployer le plug-in d'exemple.

Dans le fichier `fsumplugin_file.txt`, chaque entrée de mappage utilisateur occupe une ligne, et les mots de passe sont chiffrés par une simple inversion des octets. Il s'agit d'une méthode de chiffrement très simple. Si vous modifiez le plug-in d'exemple pour l'utiliser dans un environnement de production, vous devez utiliser une méthode de chiffrement conforme aux exigences de sécurité de l'environnement.

La syntaxe suivante montre le format pour une entrée de mappage utilisateur d'exemple.

Remarque : Bien que le code affiche plusieurs lignes ici, vous devez saisir la totalité de ce code sur une ligne dans votre application. Notez qu'un point virgule sépare les parties du mappage utilisateur et qu'un point sépare les noms d'option `REMOTE_AUTHID` et `REMOTE_PASSWORD` de leurs configurations correspondantes.

```
fsInstance;fsDatabase;fsRemoteServer;  
fsAuthid;REMOTE_AUTHID:remote_authID;  
REMOTE_PASSWORD:remote_password
```

Par exemple :

```
DB2INST1;TESTDB;MSSL2K;ALICE;REMOTE_AUTHID:TESTUSR1;REMOTE_PASSWORD:drowssap
```

Plug-in de mappage utilisateur (langage de programmation Java)

L'architecture du plug-in Java comprend deux classes d'interface et trois classes d'utilitaires. Utilisez-les pour développer un plug-in qui extrait les mappages utilisateur d'un référentiel externe.

Par défaut, les mappages utilisateur pour une source de données sont stockés localement sur chaque serveur fédéré. Un serveur LDAP stocke des objets tels que des entrées utilisateur, dans une arborescence de répertoires. Ces objets peuvent avoir des attributs tels que des mots de passe. De plus, le serveur LDAP stocke

souvent des informations supplémentaires concernant les utilisateurs, par exemple leurs adresses e-mail et leur numéro de téléphone.

La figure suivante illustre l'architecture de l'exemple de plug-in de mappage utilisateur :

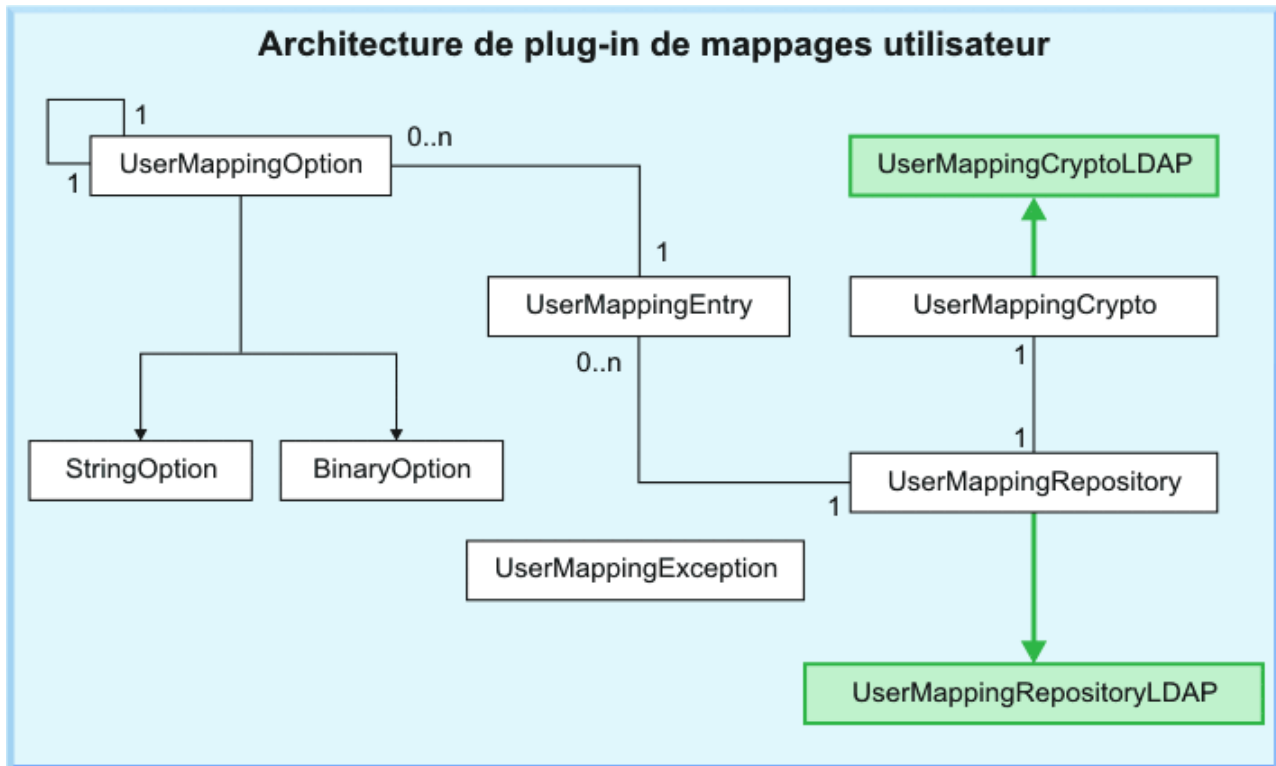


Figure 15. Architecture du plug-in de mappage utilisateur

UserMappingCrypto et UserMappingRepository sont des classes d'interface. Pour créer un plug-in de mappage utilisateur qui extrait les mappages utilisateurs du référentiel externe utilisé par votre organisation, vous devez étendre ces classes d'interface. UserMappingEntry, UserMappingOption, et UserMappingException sont des classes d'utilitaires. Vous pouvez utiliser les classes d'utilitaires sans modification. Certaines fonctions et méthodes qui sont dans les classes d'interface agissent comme fonctions de programme utilitaire. Par exemple, vous pouvez utiliser les fonctions getChars() et getBytes() dans la classe UserMappingCrypto sans modification.

Dans la figure, le terme *0..n* signifie qu'il peut exister zéro ou plusieurs de ces objets. Par exemple, un objet UserMappingEntry peut avoir plusieurs objets UserMappingOption mais il ne peut y avoir qu'un seul objet UserMappingRepository. Les classes UserMappingCryptoLDAP et UserMappingRepositoryLDAP sont étendues à partir de leurs classes parent.

Classes de plug-in de mappage utilisateur (langage de programmation Java)

L'architecture du plug-in Java comprend deux classes d'interface et trois classes d'utilitaires. Utilisez-les pour développer un plug-in qui extrait les mappages utilisateur d'un référentiel externe.

Classe UserMappingRepository (langage de programmation Java)

La classe UserMappingRepository est une classe abstraite qui n'a pas de constructeur. Pour créer votre propre plug-in de mappages utilisateur, vous devez créer une sous-classe de la classe UserMappingRepository ou modifier la sous-classe dans le plug-in d'exemple Java.

La classe UserMappingRepository contient les méthodes publiques suivantes : getVersionNumber(), getCrypto(), connect(), disconnect(), fetchUM() et lookupUM(). Vous devez créer votre propre sous-classe de la classe UserMappingRepository contenant ces fonctions. Dans ces fonctions, écrivez le code permettant au plug-in d'interagir avec votre référentiel externe.

Vous pouvez voir l'implémentation de ces fonctions dans un plug-in d'exemple Java qui extrait les mappages utilisateurs d'un serveur LDAP. Ces fichiers se trouvent dans le répertoire `sqlib/samples/federated/umplugin/ldap/`. Les fonctions de ces classes sont utilisées dans les fichiers d'exemple `UserMappingRepositoryLDAP.java` et `UserMappingLookupLDAP.java`.

Méthodes publiques

int getVersionNumber()

Renvoie le numéro de version du kit de développement de plug-in utilisé par le plug-in.

UserMappingCrypto getCrypto()

Renvoie l'objet UserMappingCrypto associé à cet objet UserMappingRepository.

abstract void connect()

Vous devez implémenter votre propre méthode de connexion à votre référentiel dans cette fonction.

abstract void disconnect()

Vous devez implémenter votre propre méthode de déconnexion de votre référentiel dans cette fonction.

abstract void fetchUM(UserMappingEntry um)

Vous devez implémenter votre propre méthode d'extraction du mappage utilisateur à partir de votre référentiel dans cette fonction. Le paramètre **um** contient les informations de requête détaillées utilisées pour déterminer le mappage utilisateur à extraire.

UserMappingEntry lookupUM(UserMappingRepository repository, String iiInstanceName, String iiDatabaseName, String iiRemoteServerName, String iiAuthid)

Cette fonction est principalement utilisée pour tester le plug-in. Cette fonction utilise les paramètres **iiInstanceName**, **iiDatabaseName**, **iiRemoteServerName** et **iiAuthid** comme entrée pour créer et initialiser la classe UserMappingEntry. Cette fonction appelle les méthodes connect, fetchUM et disconnect.

Classe UserMappingCrypto (langage de programmation Java)

Si votre référentiel externe chiffre ou code vos mots de passe distants, vous devez créer votre propre sous-classe de la classe UserMappingCrypto. Le constructeur de la sous-classe que vous créez est utilisé pour construire l'objet de cryptographie. Les méthodes de la classe de cryptographie sont appelées par d'autres classes lorsque les mots de passe de mappages utilisateur doivent être chiffrés, déchiffrés, codés ou décodés.

La classe `UserMappingCrypto` contient les méthodes publiques suivantes : `encrypt()`, `decrypt()`, `encode()` et `decode()`. Dans ces fonctions, vous devez écrire votre propre code pour chiffrer, déchiffrer, coder et décoder le mot de passe distant. Les fonctions `getBytes()` et `getChars()` sont des fonctions de programme de programme utilitaire héritées et peuvent être utilisées sans modification. Les méthodes de chiffrement, déchiffrement, codage et décodage que vous codez doivent correspondre aux méthodes de chiffrement et de codage utilisées par votre référentiel externe pour protéger les mots de passe stockés.

Vous pouvez voir l'implémentation de ces fonctions dans un plug-in d'exemple Java qui extrait les mappages utilisateurs d'un serveur LDAP. Ces fichiers se trouvent dans le répertoire `sqlib/samples/federated/umplugin/ldap/`. Les fonctions de ces classes sont utilisées dans les fichiers d'exemple `UserMappingRepositoryLDAP.java` et `UserMappingSetupLDAP.java`.

Méthodes publiques

abstract byte[] encrypt(byte[] plainValue)

Implémente l'algorithme de chiffrement correspondant à l'algorithme de chiffrement utilisé par le référentiel externe.

abstract byte[] decrypt(byte[] encryptedValue)

Implémente l'algorithme de déchiffrement inversant l'algorithme de chiffrement utilisé par votre référentiel externe puis renvoie le mot de passe.

abstract string encode(byte[] bytes)

Écrit ou implémente une fonction codant le paramètre **bytes** en chaîne. Cette fonction code la valeur chiffrée (en octets) en une chaîne.

abstract byte[] decode(String[] string)

Écrit ou implémente une fonction décodant le paramètre **string** en octets. Cette fonction est utilisée pour décoder le mot de passe extrait, sous forme de chaîne, en octets, pour que la valeur puisse être déchiffrée.

byte[] getBytes(char[] chars)

Cette fonction est héritée et peut être utilisée sans modification. Elle transforme chaque caractère d'une chaîne en octet.

char[] getChars(byte[] bytes)

Cette fonction est héritée et peut être utilisée sans modification. Elle transforme chaque octet en caractère.

Attributs protégés

SecretKey key

La clé confidentielle est utilisée pour chiffrer et déchiffrer les mots de passe distants.

Cipher cipher

L'algorithme que la clé secrète utilise pour chiffrer le mot de passe.

Classe `UserMappingEntry` (langage de programmation Java)

La classe `UserMappingEntry` est une classe utilitaire qui crée et contient les options de mappages utilisateur. Les méthodes de la classe `UserMappingEntry` sont appelées par les fonctions `fetchUM()` et `lookupUM()` de la classe `UserMappingRepository`.

Méthodes publiques

Vous pouvez voir l'utilisation de ces fonctions dans un plug-in d'exemple Java qui extrait les mappages utilisateurs d'un serveur LDAP. Ces fichiers se trouvent dans le répertoire `sqllib/samples/federated/umplugin/ldap/`. Les fonctions de ces classes sont utilisées dans les fichiers d'exemple `UserMappingRepositoryLDAP.java` et `UserMappingLookupLDAP.java`.

UserMappingEntry(UserMappingRepository **repository**, String **iiInstanceName**, String **iiDatabaseName**, String **iiRemoteServerName**, String **iiAuthID**)

Ce constructeur est utilisé pour instancier l'objet `UserMappingEntry` avec les paramètres d'entrée suivants :

- **iiInstance** - Nom d'instance du serveur fédéré
- **iiDatabase** - Nom de la base de données sur le serveur fédéré
- **iiRemoteServerName** - Nom du serveur distant pour la source de données
- **iiAuthid** - ID utilisateur local associé au mappage utilisateur

UserMappingRepository **getRepository()**

Renvoie l'objet `UserMappingRepository` associé à cet objet `UserMappingEntry`.

String **getiiInstanceName()**

Renvoie le nom de l'instance.

String **getiiDatabaseName()**

Renvoie le nom de la base de données.

String **getiiRemoteServerName()**

Renvoie le nom du serveur distant.

String **getiiAuthID()**

Renvoie le nom de l'ID utilisateur distant associé au mappage utilisateur.

UserMappingOption **getFirstOption()**

Renvoie le premier objet `UserMappingOption` appartenant à cet objet `UserMappingEntry`.

void **addOption**(UserMappingOption **newOption**)

Ajoute un nouvel objet `UserMappingOption` à cet objet `UserMappingEntry`.

Classe **UserMappingOption** (Langage de programmation Java)

La classe `UserMappingOption` est une classe d'utilitaire qui contient les fonctions fournissant au serveur fédéré l'ID utilisateur distant et le mot de passe distant.

Méthodes publiques

Vous pouvez voir l'utilisation de ces fonctions dans un plug-in d'exemple Java qui extrait les mappages utilisateurs d'un serveur LDAP. Ces fichiers se trouvent dans le répertoire `sqllib/samples/federated/umplugin/ldap/`. Les fonctions de ces classes sont utilisées dans les fichiers d'exemple `UserMappingRepositoryLDAP.java` et `UserMappingLookupLDAP.java`.

UserMappingEntry **getEntry()**

Renvoie l'objet `UserMappingEntry` associé à cet objet `UserMappingOption`.

String **getName()**

Renvoie le nom de l'option.

void **setName()**

Définit le nom de l'option.

UserMappingOption getNextOption()

Renvoie l'option suivante.

void setNextOption(UserMappingOption nextOption)

Définit l'option suivante.

abstract object getValue()

Renvoie la valeur de l'option. Cette fonction doit renvoyer une valeur de chaîne ou une valeur binaire. Reportez-vous aux méthodes StringOption et BinaryOption répertoriées ci-dessous.

classe StringOption

La classe StringOption est étendue à partir de la classe UserMappingOptions et contient les méthodes publiques suivantes : getValue() et setValue().

Méthodes publiques**object getValue()**

Renvoie la valeur de l'option de chaîne lorsque l'option est une chaîne.

void setValue(String value)

Définit la valeur de l'option de chaîne.

classe BinaryOption

La classe BinaryOption est étendue à partir de la classe UserMappingOption et contient les méthodes publiques suivantes : getValue() et setValue().

Méthodes publiques**object getValue()**

Renvoie la valeur de l'option binaire.

void setValue(byte[] value)

Définit la valeur de l'option binaire.

Classe UserMappingException (langage de programmation Java)

Le plug-in de mappage utilisateur Java utilise la classe UserMappingException, qui est une sous-classe de la classe java.lang.Exception, pour signaler des erreurs.

Méthodes publiques

Vous pouvez voir l'utilisation de ces fonctions dans un plug-in d'exemple Java qui extrait les mappages utilisateurs d'un serveur LDAP. Ces fichiers se trouvent dans le répertoire sqllib/samples/federated/umplugin/ldap/. Les fonctions de cette classe sont utilisées dans chaque fichier Java d'exemple.

UserMappingException(int errorNumber)

Constructeur utilisé pour l'instanciation de l'objet UserMappingException utiliser pour rapporter les erreurs. Le paramètre **errorNumber** envoyé à l'objet UserMappingException définit le type d'erreur rapportée.

int getErrorNumber()

Renvoie le numéro d'erreur de l'exception.

Le fichier UserMappingRepositoryLDAP.java contient cette fonction pour intercepter et rapporter les erreurs de test d'un plug-in LDAP.

String getErrorMessage()

Renvoie le message d'erreur de l'exception.

Le fichier `UserMappingRepositoryLDAP.java` contient une méthode pour intercepter et rapporter les erreurs de test d'un plug-in LDAP.

Messages d'erreur

Les numéros d'erreurs, noms de constantes et messages d'erreur sont répertoriés dans la table suivante.

Tableau 26. Numéros et messages d'erreur

Numéro d'erreur	Nom de la constante	Message d'erreur
1	INITIALIZE_ERROR	Echec de l'initialisation du plug-in.
2	CONNECTION_ERROR	Connexion au référentiel impossible.
3	AUTHENTICATION_ERROR	Authentification avec le référentiel impossible.
4	LOOKUP_ERROR	Echec de la recherche sur le référentiel.
5	DECRYPTION_ERROR	Echec du déchiffrement.
6	DISCONNECT_ERROR	Déconnexion du référentiel impossible.
7	INVALID_PARAMETER_ERROR	Paramètre incorrect.
8	UNAUTHORIZED_CALLER	L'appelant n'est pas autorisé à appeler le plug-in.

Plug-in exemple de mappage utilisateur (langage de programmation Java)

Le plug-in d'exemple Java extrait les entrées de mappage utilisateur d'un serveur LDAP. Pour utiliser le plug-in dans votre environnement, modifiez le plug-in d'exemple afin qu'il corresponde aux paramètres utilisés par votre serveur LDAP.

Les fichiers pour le plug-in se trouvent dans le répertoire `sqlib/samples/federated/umplugin/ldap/`. Le tableau suivant décrit chaque fichier. Avant de modifier les fichiers, copiez-les dans un répertoire de travail vide. Puis travaillez avec les copies.

Tableau 27. Description des fichiers dans le plug-in d'exemple de mappage utilisateur (Java)

Nom du fichier	Description
<code>README.txt</code>	Ce fichier contient une version condensée des instructions et de la documentation relatives au test et à l'utilisation du plug-in d'exemple.
<code>UserMappingCryptoLDAP.java</code>	Cette classe Java contient le code qui implémente les mesures de sécurité pour le chiffrement, le déchiffrement, le codage et le décodage des mappages utilisateur extrait du serveur LDAP. Vous devez modifier ce fichier afin qu'il fonctionne avec votre serveur LDAP.
<code>UserMappingSetupLDAP.java</code>	Cette classe Java crée le fichier de configuration qui stocke les informations de connexion LDAP ainsi que d'autres paramètres de configuration, y compris l'adresse IP ou le nom d'hôte, SSL ou non-SSL, ID utilisateur et mot de passe.

Tableau 27. Description des fichiers dans le plug-in d'exemple de mappage utilisateur (Java) (suite)

Nom du fichier	Description
UserMappingRepositoryLDAP.java	Cette classe Java contient le code pour la connexion, la déconnexion et l'extraction des mappages utilisateur du serveur LDAP. Le code de ce fichier utilise le schéma défini dans le fichier schema.ldif. Si vous voulez modifier le schéma, vous devez également modifier une section de ce fichier.
UserMappingLookupLDAP.java	Cette classe Java contient le code pour effectuer un test de recherche LDAP. Utilisez ce fichier pour tester votre plug-in en dehors du serveur fédéré. Puis testez le plug-in sur le serveur fédéré.
schema.ldif	Ce fichier est chargé sur votre serveur LDAP afin de définir le schéma. Le fichier LDIF (Lightweight Directory Interchange Format) contient des objets et des attributs qui sont ajoutés au serveur LDAP.
entry.ldif	Le fichier entry.ldif ajoute des entrées au serveur LDAP. Les entrées utilisateur utilisent les objets et attributs du fichierschema.ldif pour stocker les mappages utilisateur.

Développement d'un plug-in de mappage utilisateur (langage de programmation Java)

Utilisez le code échantillon comme point de départ pour développer un plug-in Java qui extrait les mappages utilisateur d'un référentiel externe. Le code échantillon extrait les mappages d'un référentiel LDAP, mais vous pouvez modifier le code afin d'accéder à tout référentiel externe.

Vérifiez les éléments suivants :

- Le Kit de Développement Java (JDK) version 1.4 ou ultérieure est installé.
- Le fichier db2umplugin.jar. Ce fichier Java Archive (JAR) est installé dans le cadre de l'installation du serveur DB2 ou de l'installation du client DB2.
- Les fichiers de plug-in de mappage utilisateur échantillons sont installés. ces fichiers qui sont installés dans le cadre de l'installation client DB2 sont dans le répertoire sqllib/samples/federated/umplugin/ldap/.
- Le paramètre `java_heap_sz` est défini à 2048.

Le plug-in que vous développez doit pouvoir se connecter au référentiel externe, récupérer les mappages utilisateur, et décrypter les mots de passe distants. Le référentiel que vous utilisez détermine la manière dont vous codez le plug-in. Par exemple, si vous utilisez un référentiel LDAP qui stocke les mots de passe chiffrés, le plug-in doit contenir le schéma de chiffrement et la clé secrète nécessaires pour décoder les mots de passe.

Sachez qu'à mesure que vous développez et utilisez le plug-in, vous envoyez des mots de passe et ID utilisateur sensibles entre plusieurs sources. Pour protéger ces informations, restreignez l'accès au code source du plug-in, et configurez l'utilitaire de contrôle de db2audit pour capturer un enregistrement VALIDATE dans le fichier journal de diagnostic db2diag.log, chaque fois que le serveur fédéré utilise le plug-in. Le fichier journal de diagnostic est utile non seulement pour tracer l'utilisation mais aussi pour le diagnostic de panne de tout incident qui se produit.

Pour développer un plug-in, procédez comme suit :

Modification des fichiers du plug-in d'exemple de mappage utilisateur (langage de programmation Java)

Le plug-in d'exemple Java extrait les mappages utilisateur d'un serveur LDAP. Vous pouvez modifier le plug-in pour extraire des mappages utilisateurs à partir d'autres types de référentiels externes. Utilisez le plug-in d'exemple comme point de départ pour développer un plug-in personnalisé fonctionnant avec votre référentiel externe spécifique.

Chacun des fichiers d'exemple effectue une tâche différente du processus d'extraction des mappages utilisateur. La plupart des fonctions et des classes du code d'exemple doivent être modifiées pour fonctionner avec votre référentiel externe spécifique. Le tableau suivant décrit les fonctions importantes.

Tableau 28. Fonctions et classes à modifier

Nom du fichier	Élément à modifier	Classe à référencer
UserMappingCryptoLDAP.java	UserMappingCryptoLDAP() encrypt() decrypt() getKey() decode() encode()	Classe UserMappingCrypto Classe UserMappingException
UserMappingRepositoryLDAP.java	classe UserMappingRepositoryLDAP(String configFilePath) connect() disconnect() fetchUM()	Classe UserMappingRepository Classe UserMappingEntry Classe UserMappingOption Classe UserMappingException
UserMappingSetupLDAP.java	Vous devez modifier ce fichier pour créer un fichier de configuration qui stocke les valeurs dont votre plug-in a besoin pour la connexion et l'extraction de mappages utilisateur à partir du référentiel externe. Si vous créez manuellement un fichier de configuration, vérifiez que tous les mots de passe que vous stockez dans ce fichier sont chiffrés. Le nom du fichier de configuration doit correspondre au nom de la classe de référentiel. Par exemple, la classe UserMappingRepositoryXXXX et le fichierUserMappingRepositoryXXXX.cfg.	

Modification du fichier d'exemple UserMappingCryptoLDAP (langage de programmation Java)

Pour implémenter les méthodes de sécurité qu'utilise le serveur LDAP, modifiez les fonctions qui chiffrent, déchiffrent, codent et décotent les mots de passe distants.

Dans la mesure où les méthodes de chiffrement doivent être secrètes et uniques, cette tâche spécifie les sections et les fonctions à modifier. Vous personnalisez le code pour implémenter les méthodes de sécurité utilisées par votre serveur LDAP.

Pour modifier les fonctions de sécurité dans le fichier UserMappingCryptoLDAP :

1. Ouvrez le fichier UserMappingCryptoLDAP.java avec un éditeur de texte.
2. Sous les mentions de copyright et la clause de protection IBM, importez les packages auxquels votre code fera référence. Le plug-in d'exemple utilise les packages javax.crypto et javax.crypto.spec Java, qui donnent les classes pour le code de déchiffrement (chiffrement et déchiffrement), ainsi que les paramètres d'algorithme et de clé. Remplacez ces packages Java par votre propre package.

3. Mettez à jour les fonctions suivantes :

public UserMappingCryptoLDAP()

Remplacez le code de chiffrement par celui du code de chiffrement correspondant au chiffrement de mot de passe utilisé par votre serveur LDAP.

public byte[] encrypt(byte[] plainValue)

Cette fonction fournit le code qui chiffre les mots de passe afin qu'ils puissent être stockés sur le serveur LDAP. Cette fonction chiffre également le mot de passe de connexion LDAP stocké dans le fichier de configuration.

Remplacez le code de cette fonction par votre propre code chiffrant le paramètre **plainValue**.

public byte[] decrypt(byte[] encryptedValue)

Remplacez le code de cette fonction par votre propre code déchiffrant le paramètre **encryptedValue**.

private SecretKey getKey()

Remplacez le code de cette fonction par le code utilisé pour fournir au plug-in la clé permettant de chiffrer et déchiffrer vos mots de passe.

public byte[] decode(String string)

Les mots de passe sont d'abord chiffrés puis codés. Cette fonction fournit le code pour décoder les mots de passe avant que ceux-ci ne soient déchiffrés. Les mots de passe chiffrés sont codés afin de transformer la sortie binaire du mot de passe chiffré en caractères ASCII.

Remplacez le code de cette fonction par votre propre code décodant le paramètre **string**.

public String encode(byte[] bytes)

Les mots de passe sont d'abord chiffrés puis codés. Cette fonction fournit le code pour coder la sortie binaire des mots de passe chiffrés. Les mots de passe chiffrés sont codés afin de transformer la sortie binaire du mot de passe chiffré en caractères ASCII.

Remplacez le code de cette fonction par votre propre code encodant le paramètre **bytes**.

Modification du fichier d'exemple UserMappingRepositoryLDAP (langage de programmation Java)

Le fichier `UserMappingRepositoryLDAP.java` pour le plug-in Java contient les fonctions de connexion au serveur LDAP, d'extraction des mappages utilisateur et de déconnexion. Vous devez modifier le code d'exemple pour qu'ils correspondent aux classes d'objets définies dans la structure de répertoire de votre serveur LDAP.

Pour extraire les mappages utilisateur à partir du serveur LDAP, le plug-in doit rechercher dans le répertoire les entrées utilisateur comportant les attributs définissant l'entrée qui définissent le mappage utilisateur. Les informations suivantes sont stockées en tant qu'attributs d'un d'utilisateur :

- Remote server name
- Nom d'instance
- Nom de base de données
- Nom d'utilisateur distant
- Mot de passe d'utilisateur distant

Le code d'exemple considère que l'entrée utilisateur sera identifiée par la classe d'objets *inetOrgPerson* et que l'entrée de mappage utilisateur sera identifiée par la classe d'objets *IIUserMapping*. Les fichiers d'exemple LDIF (Lightweight Directory Interchange Format), *schema.ldif* et *entry.ldif*, sont utilisés pour charger le schéma et les entrées d'exemples sur le serveur LDAP.

Le code du fichier *UserMappingRepositoryLDAP.java* considère que le fichier *schema.ldif* contient le code LDIF pour les objets et les attributs ayant les noms suivants :

- Objet *IIUserMapping*
 - Attribut *IIRemoteServerName*
 - Attribut *IIInstanceName*
 - Attribut *IIDatabaseName*
 - Attribut *IIRemotePassword*
 - Attribut *uid*

Vous devez modifier le fichier *UserMappingRepositoryLDAP.java* pour qu'il corresponde au schéma de votre serveur LDAP. Le fichier *UserMappingRepositoryLDAP.java* doit rechercher et extraire les entrées de mappage utilisateur du serveur LDAP. Les fichiers LDIF sont fournis en tant que schéma et méthode d'exemple pour le stockage des entrées de mappage utilisateur.

Pour modifier le schéma utilisé par le fichier *UserMappingRepositoryLDAP.java* :

1. Recherchez le code : `private String UserObjectClassName = "inetOrgPerson"` et remplacez la valeur *inetOrgPerson* par le nom de la classe d'objets pour les entrées utilisateur qu'utilise votre serveur LDAP.
2. Facultatif : Modifiez les noms d'attributs utilisés par le plug-in. Remplacez les valeurs des variables *IIRemoteServerAttrName*, *IIInstanceAttrName*, *IIDatabaseAttrName* et *IIRemotePasswordAttrName* par les noms d'attributs que vous avez choisis.
3. Si vous choisissez d'utiliser les fichiers LDIF, vérifiez que le schéma de ces fichiers correspond à la structure recherchée par le fichier *UserMappingRepositoryLDAP.java*.

Compilation des fichiers pour le plug-in de mappage utilisateur (langage de programmation Java)

Après avoir modifié les fichiers source du plug-in Java, vous devez les compiler.

Dans les commandes ci-dessous, si votre chemin d'accès complet comporte des espaces, vous devez l'entourer de guillemets. Par exemple, "C:\program files\sqlib\java\db2umplugin.jar". *%DB2PATH%* est le répertoire dans lequel DB2 est installé, par exemple C:\ProgramFiles\IBM\sqlib. *inst_home* est le répertoire initial d'instance.

Les commandes ci-dessous considèrent que vous utilisez la convention de dénomination basée sur le nom des classes. Remplacez *xxxx* par le nom choisi.

Pour compiler les fichiers source :

1. Emettez la commande de compilation :

Windows :

```
javac -classpath "%DB2PATH%\java\db2umplugin.jar; ^
%CLASSPATH%" -d . ^
.\UserMappingRepositoryxxxx.java ^
.\UserMappingCryptoxxxx.java ^
.\UserMappingSetupxxxx.java ^
.\UserMappingLookupXXXX.java
```

UNIX :

```
javac -classpath inst_home/sqlllib/java/db2umplugin.jar:\
$CLASSPATH -d . \
./UserMappingRepositoryxxxx.java \
./UserMappingCryptoxxxx.java \
./UserMappingSetupxxxx.java \
./UserMappingLookupxxxx.java
```

2. Archivez les fichiers de classe Java dans un seul fichier JAR (Java Archive) : Le point (.) situé après le nom du fichier de sortie indique à la commande de rechercher et placer les fichiers dans le même répertoire. Si vous modifiez le répertoire, utilisez le chemin de fichier approprié, selon votre système d'exploitation. Par exemple, /home/user/folder ou C:\test\folder).

```
jar -cfM0 UserMappingRepositoryXXXX.jar .
```

Création du fichier de configuration pour le plug-in de mappage utilisateur (langage de programmation Java)

Le fichier de configuration stocke les informations de connexion que le plug-in Java utilise pour se connecter au serveur LDAP.

Pour créer le fichier de configuration, exécutez le programme de configuration qui vous invite à fournir les informations suivantes :

- Nom d'hôte ou adresse IP du serveur LDAP
- Numéro de port du serveur LDAP (la valeur par défaut est 389)
- Nom distinctif de la branche LDAP (par exemple, ou=ii,o=ibm,c=us)
- ID utilisateur pour la connexion au serveur LDAP
- Mot de passe pour la connexion au serveur LDAP
- Configuration SSL

L'entrée que vous fournissez est utilisée pour créer le fichier `UserMappingRepositoryLDAP.cfg`, qui stocke les informations de configuration. Si un mot de passe est requis pour la connexion à votre serveur LDAP, il le mot de passe est chiffré avec l'algorithme que vous avez spécifié dans le fichier `UserMappingCryptoLDAP.java` .

Dans les commandes ci-dessous, si votre chemin d'accès complet comporte des espaces, vous devez l'entourer de guillemets. Par exemple, "C:\program files\sqllib\java\db2umplugin.jar". `%DB2PATH%` est le répertoire dans lequel DB2 est installé, par exemple C:\ProgramFiles\IBM\sqllib. `inst_home` est le répertoire initial d'instance.

Pour créer le fichier de configuration pour le plug-in LDAP d'exemple :

Windows :

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^
.\UserMappingRepositoryLDAP.jar;%CLASSPATH%" UserMappingSetupLDAP
```

UNIX :

```
java -classpath inst_home/sqllib/java/db2umplugin.jar: \
./UserMappingRepositoryLDAP.jar:$CLASSPATH UserMappingSetupLDAP
```

Test du plug-in de mappages utilisateur (langage de programmation Java)

Pour tester le plug-in Java en dehors du serveur fédéré, développez une application qui se connecte à et extrait les mappages utilisateurs du référentiel externe.

Vous pouvez développer un programme simple qui appelle la méthode `lookupUM()` dont votre classe `XXXX UserMappingRepository` a hérité de la classe `UserMappingRepository`, pour la connexion et l'extraction de mappages utilisateur de votre référentiel externe. Vous pouvez afficher le fichier `UserMappingLookupLDAP.java` qui se trouve dans le répertoire `sqllib/samples/federated/umplugin/ldap/`.

Dans les commandes ci-dessous, si votre chemin d'accès complet comporte des espaces, vous devez l'entourer de guillemets. Par exemple, "`C:\program files\sqllib\java\db2umplugin.jar`". `%DB2PATH%` est le répertoire dans lequel DB2 est installé, par exemple `C:\ProgramFiles\IBM\sqllib`. `inst_home` est le répertoire initial d'instance.

Le programme de test doit utiliser les paramètres requis pour l'identification du mappage utilisateur :

- *remoteServerName* - Nom du serveur distant pour la source de données
- *iiAuthid* - ID utilisateur local associé au mappage utilisateur
- *iiInstance* - Nom d'instance du serveur fédéré
- *iiDatabase* - Nom de la base de données sur le serveur fédéré

Pour tester le plug-in de mappages utilisateur :

Windows :

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^
.\UserMappingRepositoryXXXX.jar;%CLASSPATH%" ^
UserMappingLookupXXXX -server remoteServerName ^
-authid iiAuthid ^
-instance iiInstance -database iiDatabase
```

UNIX :

```
java -classpath inst_home/sqllib/java/db2umplugin.jar: ^
./UserMappingRepositoryXXXX.jar:$CLASSPATH \
UserMappingLookupXXXX -server remoteServerName \
-authid iiAuthid \
-instance iiInstance -database iiDatabase
```

Déploiement des fichiers du plug-in de mappage utilisateur (langage de programmation Java)

Après avoir compilé et testé le plug-in Java, déployez les fichiers sur le serveur fédéré.

Dans les commandes ci-dessous, *sqllib* est le chemin d'accès complet à votre installation DB2. Les commandes ci-dessous considèrent que vous utilisez la convention de dénomination basée sur le nom des classes. Remplacez *xxxx* par le nom choisi.

Pour déployer les fichiers compilés :

Copiez vos fichiers `UserMappingRepositoryxxxx.jar` et `UserMappingRepositoryxxxx.cfg` dans le répertoire `sqllib/function/`.

En plaçant les fichiers dans ce répertoire, le serveur fédéré peut charger et appeler les classes contenues dans ces fichiers.

Configuration de l'accès au plug-in de mappage utilisateur (langage de programmation Java)

Vous devez définir l'option `DB2_UM_PLUGIN` pour utiliser un plug-in Java pour l'accès aux mappages utilisateur dans le référentiel externe.

Avant de commencer

Avant de configurer le serveur fédéré pour accéder aux mappages utilisateur dans un référentiel externe, vous devez effectuer l'une des tâches suivantes :

- Développer un plug-in de mappages utilisateur
- Déployer le plug-in sur le serveur fédéré
- Mettre à jour la configuration de votre gestionnaire de bases de données :

```
db2 update dbm cfg using JDK_PATH votre_chemin_jdk
db2 terminate
db2stop
db2start
```

L'option `DB2_UM_PLUGIN` doit contenir le chemin d'accès complet de la classe, y compris le nom du package. Si vous développez un package, vous devez inclure son nom avant le nom de la classe. Par exemple, `'package.nomclasse'`. Si vous utilisez le plug-in d'exemple LDAP, spécifiez la valeur `'UserMappingRepositoryLDAP'` dans l'option `DB2_UM_PLUGIN`. Le plug-in d'exemple n'est pas développé en tant que package.

Pour configurer le serveur fédéré pour accéder au référentiel externe :

Choisissez la méthode d'implémentation du plug-in de mappages utilisateur :

Méthode	Instruction SQL
Spécifiez le plug-in de mappages utilisateur lors de la création d'un encapsuleur	<pre>CREATE WRAPPER <i>wrapper-name</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>
Modifiez un encapsuleur existant pour spécifier le plug-in de mappages utilisateur	<pre>ALTER WRAPPER <i>wrapper-name</i> (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>
Spécifiez le plug-in de mappages utilisateur lors de la création d'une définition de serveur	<pre>CREATE SERVER <i>server_name</i> TYPE <i>date_source_type</i> VERSION <i>version_number</i> WRAPPER <i>wrapper-name</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>
Modifiez une définition de serveur existante pour spécifier le plug-in de mappages utilisateur	<pre>ALTER SERVER <i>server_name</i> OPTIONS (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>

Une fois définie l'option DB2_UM_PLUGIN, le serveur fédéré utilise les informations de connexion spécifiées dans le fichier UserMappingRepositoryXXXX.cfg pour extraire les mappages utilisateur du référentiel externe. XXXX est le nom du plug-in.

Pour modifier l'encapsuleur afin d'utiliser un autre plug-in

```
ALTER WRAPPER wrapper-name (
  SET DB2_UM_PLUGIN 'com.package_name.um.UserMappingRepositoryXXXX'
);
```

Pour modifier une définition de serveur afin d'utiliser un autre plug-in

```
ALTER SERVER server_name
  OPTIONS (
    SET DB2_UM_PLUGIN 'UserMappingRepositoryXXXX'
  );
```

Chapitre 34. Sécurité Oracle dans un système fédéré

La fédération prend en charge un ensemble de fonctions de sécurité Oracle.

Si vous implémentez ces fonctions dans Oracle, le serveur fédéré peut en profiter.

Oracle Label Security

Le serveur fédéré prend en charge Oracle Label Security, qui vous permet de sécuriser des données et de garantir que seuls les utilisateurs possédant les droits nécessaires peuvent les voir.

Les administrateurs peuvent utiliser Oracle Label Security pour appliquer des règles de sécurité à chaque ligne d'une table. Les règles de sécurité déterminent un niveau d'accès aux données d'un utilisateur en fonction des droits accordés à l'ID utilisateur ou à l'ID de la session. Lorsque vous créez un pseudonyme sur un objet de source de données Oracle, le serveur fédéré détermine automatiquement si la source de données utilise Oracle Label Security. Si Oracle Label Sécurité est utilisé, le pseudonyme n'est pas mis en mémoire cache.

Vous pouvez utiliser l'instruction ALTER NICKNAME pour autoriser ou non la mise en mémoire cache. Par exemple, si vous avez créé un pseudonyme sur un objet de source de données avec Oracle Label Security avant que la prise en charge fédérée de cette fonction ne soit disponible, vous pouvez modifier le pseudonyme pour refuser la mise en mémoire cache. Si vous avez créé un pseudonyme sur un objet de source de données avec Oracle Label Security et si Oracle Label Security est supprimé, vous pouvez modifier le pseudonyme pour autoriser la mise en mémoire cache.

Un administrateur de base de données peut choisir de masquer un intitulé afin d'éviter que certains utilisateurs ne sachent qu'une ligne donnée existe. Dans ce cas, la colonne est masquée dans la table. Les pseudonymes avec une colonne d'intitulé masquée ne sont pas mis en mémoire cache.

Authentification de proxy Oracle et contextes accrédités fédérés

Créez une connexion physique vers la source de données Oracle, puis basculez la connexion vers un utilisateur différent sur la même connexion.

L'utilisation de l'authentification de proxy Oracle et de contextes accrédités fédérés permet de réduire le temps système du réseau pour la création d'une connexion réseau séparée du serveur fédéré vers la base de données Oracle pour chaque utilisateur, tout en affirmant positivement l'identité de l'utilisateur connecté à la source de données Oracle. L'application peut passer d'un utilisateur à l'autre, si nécessaire, pour traiter des transactions de la part des utilisateurs.

Pour configurer ce scénario, procédez comme suit :

1. Sur le serveur Oracle, utilisez l'instruction Oracle ALTER USER pour enregistrer chaque proxy utilisateur. Ici, le droit d'utiliser le proxy nommé BOSS est accordé à Mary, qui est l'utilisateur de proxy et qui obtient le rôle CLERK pour la durée de la connexion :

```
ALTER USER MARY GRANT  
CONNECT THROUGH BOSS  
WITH ROLE CLERK
```

2. Sur le serveur fédéré, créez cet objet de contexte accrédité :

```
CREATE TRUSTED CONTEXT MY_FED_TCX  
BASED UPON CONNECTION USING SYSTEM AUTHID BOSS  
ATTRIBUTES (ENCRYPTION 'NONE')  
WITH USE FOR MARY WITHOUT AUTHENTICATION  
ENABLE
```

Avec cette configuration, le serveur fédéré peut installer une connexion accréditée de bout en bout depuis le client vers la source de données Oracle via le serveur fédéré. Le BOSS peut établir une connexion accréditée et MARY peut la réutiliser.

Pour établir et réutiliser des connexions accréditées, utilisez l'interface de programme d'application fournie par DB2.

Chapitre 35. Support de source de données pour des fonctions fédérées

Reportez-vous à ce tableau pour savoir si une source de données prend ou non en charge une fonction fédérée spécifique.

Avant de pouvoir utiliser certaines de ces fonctions, vous devrez éventuellement définir des options d'encapsuleur ou de serveur spécifiques, ou exécuter d'autres tâches pour les activer. Pour plus d'informations, reportez-vous aux rubriques spécifiques à chaque fonction.

Tableau 29. Fonctions et sources de données prises en charge

Fonction	Sources de données
Points de sauvegarde d'application avec opérations WRITE sur les pseudonymes	DB2 for Linux, UNIX et Windows
Optimisation de l'état asynchrone	Toutes les sources de données
Tables de la mémoire cache	famille DB2 Informix Microsoft SQL Server Oracle Sybase
Importation de données dans des pseudonymes	famille DB2 Informix Microsoft SQL Server Oracle Sybase Teradata
Tolérances des erreurs dans les expressions de tables imbriquées	famille DB2 Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Teradata
Référentiel de mappage utilisateur externe	Toutes les sources de données
Indicateurs d'état de santé fédérés	famille DB2 Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Fichiers structurés en tableaux Teradata XML (uniquement pour les pseudonymes racines)

Tableau 29. Fonctions et sources de données prises en charge (suite)

Fonction	Sources de données
Procédures fédérées	famille DB2, en mode sécurisé Oracle, en mode sécurisé Microsoft SQL Server, en mode sécurisé Sybase, en mode isolé, avec le serveur fédéré installé sous UNIX Sybase, en mode isolé ou mode sécurisé, avec le serveur fédéré installé sous Linux ou Microsoft Windows
Contextes accrédités fédérés	DB2 for Linux, UNIX et Windows version 9.5 DB2 for z/OS version 9 Oracle
Proxy HTTP	Services Web XML
Isolement au niveau de la connexion	famille DB2 Informix JDBC Microsoft SQL Server ODBC Oracle Sybase
Contrôle d'accès basé sur les labels	DB2 for Linux, UNIX et Windows version 9.1 et 9.5 Oracle
Opérations LOB en lecture et écriture	DB2 for z/OS DB2 for Linux, UNIX et Windows DB2 for System i Oracle
Opérations LOB en lecture seule	BioRS Informix JDBC Microsoft SQL Server ODBC Script Sybase Teradata Services Web XML
Tables de requêtes matérialisées	Toutes les sources de données, avec des restrictions spécifiques
Fonction de mise à jour des statistiques de pseudonyme	BioRS famille DB2 Excel Informix JDBC Microsoft SQL Server ODBC Oracle Sybase Fichiers structurés en tableaux Teradata XML (uniquement pour les pseudonymes racines)

Tableau 29. Fonctions et sources de données prises en charge (suite)

Fonction	Sources de données
Sessions passe-système	DRDA Informix Oracle Microsoft SQL Server Sybase Teradata
Type de données XML distant	DB2 for Linux, UNIX et Windows encapsuleur XML
Proxy SOCKS	BioRS Script Services Web XML
Secure Socket Layer (SSL)	Services Web XML
Isolement au niveau de l'instruction	famille DB2 Microsoft SQL Server
Transactions de validation en deux phases	DB2 for Linux, UNIX et Windows, en mode sécurisé DB2 for System i, en mode sécurisé DB2 for z/OS, en mode sécurisé Informix, en mode sécurisé Microsoft SQL Server, en mode sécurisé, avec le serveur fédéré installé sous Microsoft Windows Oracle, en mode sécurisé Sybase, en mode sécurisé, avec le serveur fédéré installé sous Microsoft Windows Sybase, en mode isolé, avec le serveur fédéré installé sous UNIX
Prise en charge de Unicode	Toutes les sources de données

Chapitre 36. Référence des options des sources de données

Chaque source de données prend en charge des options d'encapsuleur, de serveur, de mappage utilisateur, de pseudonyme et de colonne spécifiques.

Référence des options BioRS

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier dans les instructions CREATE WRAPPER et CREATE SERVER.

Tableau 30. Options d'encapsuleur pour BioRS

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Cette option est obligatoire si la valeur PROXY_TYPE est HTTP ou SOCKS. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Cette option est obligatoire si la valeur PROXY_TYPE est HTTP ou SOCKS. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.

Tableau 30. Options d'encapsuleur pour BioRS (suite)

Nom	Description
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE. Si vous définissez cette option sur HTTP ou SOCKS, vous devez également spécifier PROXY_SERVER_NAME et PROXY_SERVER_PORT.

Options de serveur

Tableau 31. Options de serveur pour BioRS

Nom	Description
CASE_SENSITIVE	Spécifie si le serveur BioRS traite les noms en étant sensible à la casse. Les valeurs valides sont Y et N. La valeur par défaut est Y ; les noms sont traités avec sensibilité à la casse. Dans le produit BioRS, un paramètre de configuration contrôle la sensibilité à la casse des données stockées sur le serveur BioRS. L'option CASE_SENSITIVE et le paramètre de configuration doivent spécifier la même sensibilité à la casse. Si vous avez besoin de modifier la valeur de l'option CASE_SENSITIVE après avoir créé la définition de serveur, vous devez supprimer la définition de serveur et créer à nouveau la définition et tous les pseudonymes.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 31. Options de serveur pour BioRS (suite)

Nom	Description
NODE	Obligatoire. Spécifie le nom d'hôte DNS ou l'adresse IP du système sur lequel l'utilitaire d'interrogation BioRS est disponible. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré. La valeur par défaut est localhost.
PORT	Spécifie le port des connexions vers le serveur BioRS. Les valeurs valides sont un port numérique ou un nom de service TCP/IP. La valeur par défaut est 5014.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
TIMEOUT	Spécifie la durée maximale, en minutes, au cours de laquelle le serveur fédéré attend une réponse du serveur distant. La valeur par défaut est 10.

Options de mappage utilisateur

Tableau 32. Options de mappage utilisateur pour BioRS

Option	Description
GUEST	Spécifie que l'ID d'authentification BioRS GUEST est utilisée pour effectuer des opérations. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'ID BioRS GUEST n'est pas utilisé. Cette option n'est pas valide si vous spécifiez les options REMOTE_AUTHID et REMOTE_PASSWORD.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.

Tableau 32. Options de mappage utilisateur pour BioRS (suite)

Option	Description
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de pseudonyme

Tableau 33. Options de pseudonyme pour BioRS

Option	Description
REMOTE_OBJECT	Spécifie le nom de la banque de données BioRS associée au pseudonyme. Ce nom détermine le schéma et la banque de données BioRS pour le pseudonyme. Le nom spécifie également la relation du pseudonyme avec les autres pseudonymes. La sensibilité à la casse de cette option dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Vous ne pouvez pas utiliser l'instruction ALTER NICKNAME pour modifier ou supprimer ce nom. Si le nom d'une banque de données BioRS change, vous devez supprimer le pseudonyme et le recréer.
TIMEOUT	Spécifie la durée d'attente maximale, en minutes, d'une réponse provenant du serveur de source de données. La valeur par défaut est 10.

Options de colonne

Tableau 34. Options de colonne pour BioRS

Option	Description
ELEMENT_NAME	Spécifie le nom de l'élément BioRS. La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Vous devez spécifier le nom de l'élément BioRS uniquement si il est différent du nom de colonne.

Tableau 34. Options de colonne pour BioRS (suite)

Option	Description
IS_INDEXED	Spécifie si la colonne correspondante est indexée et, par conséquent, si elle peut être référencée dans un prédicat. Les valeurs valides sont Y et N. La valeur par défaut est N. La colonne n'est pas indexée.
REFERENCED_OBJECT	Spécifie le nom de la banque de données BioRS référencée par la colonne en cours. La sensibilité à la casse de ce nom dépend de la sensibilité à la casse du serveur BioRS et de la valeur de l'option du serveur CASE_SENSITIVE. Cette option n'est valide que pour les colonnes ayant le type de données de référence BioRS.

Référence des options de base de données DB2

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent aux sources de données DB2 et identifient les options obligatoires que vous devez spécifier.

Tableau 35. Options d'encapsuleur pour les sources de données DB2

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 36. Options de serveur pour les sources de données DB2

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DATE_COMPAT	Indique si le paramètre date_compat est appliqué à la base de données. Les valeurs valides sont Y et N. La valeur par défaut est N. Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows, version 9.7 ou ultérieure.
DBNAME	Obligatoire. Indique la base de données spécifique à utiliser pour la connexion initiale à la base de données DB2 distante. Cette base de données spécifique représente l'alias de base de données pour la base de données DB2 distante qui est cataloguée sur le serveur fédéré à l'aide de la commande CATALOG DATABASE ou de l'Assistant Configuration de DB2.

Tableau 36. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant les options REMOTE_AUTHID et REMOTE_PASSWORD. Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS version 9 et ultérieure.

Tableau 36. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NO_EMPTY_STRING	Indique si le serveur source de données distant peut contenir des chaînes vides. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données distantes Oracle, la valeur par défaut est Y ; toutes les valeurs de chaîne vides sont converties en valeurs NULL. Pour toutes les autres sources de données distantes, la valeur par défaut est N ; la source de données peut contenir des chaînes vides. Vous pouvez améliorer les performances de vos systèmes en définissant cette option sur Y dans les configurations système où le serveur fédéré est en mode compatible VARCHAR2, mais où la source de données distante n'est pas compatible VARCHAR2.

Tableau 36. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
NUMBER_COMPAT	<p>Indique si le serveur de source de données prend en charge le type de données NUMBER. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur de source de données ne prend pas en charge le type de données NUMBER. Dans les systèmes où le serveur fédéré ne prend pas en charge le type de données NUMBER contrairement au serveur de source de données, vous devez définir l'option NUMBER_COMPAT sur Y, car le serveur de source de données peut renvoyer des résultats DECFLOAT hors de l'intervalle du type de données DECIMAL et générer l'erreur SQLSTATE 560BD.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.7 et ultérieure.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>
PUSHDOWN	<p>Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.</p>

Tableau 36. Options de serveur pour les sources de données DB2 (suite)

Nom	Description
SAME_DECFLT_ROUNDING	<p>Indique si le mode d'arrondissement du serveur fédéré et du serveur de source de données utilise les mêmes paramètres de mode d'arrondissement DECFLOAT. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré et le serveur distant ont les mêmes paramètres de mode d'arrondissement DECFLOAT.</p> <p>Important : Si vous définissez cette option sur Y lorsque les modes d'arrondissement diffèrent entre le serveur fédéré et le serveur de source de données, il est possible que vous obteniez des résultats d'arrondissement DECFLOAT incorrects.</p> <p>Pour configurer un serveur fédéré et un serveur de source de données existants possédant le même paramètre d'arrondissement DECFLOAT, utilisez l'instruction ALTER SERVER.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure.</p>
VARCHAR2_COMPAT	<p>Indique si la source de données distante est compatible VARCHAR2. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données Oracle, la valeur par défaut est Y ; la source de données est compatible VARCHAR2. Pour toutes les sources de données distantes, la valeur par défaut est N ; la source de données n'est pas définie sur le mode compatible VARCHAR2.</p> <p>Vous devez définir cette option de serveur sur Y si la source de données DB2 Database pour Linux, UNIX et Windows, ODBC ou JDBC est configurée en mode compatible VARCHAR2.</p>

Options de mappage utilisateur

Tableau 37. Options de mappage utilisateur pour les sources de données DB2

Option	Description
FED_PROXY_USER	<p>Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD. Si vous spécifiez l'option de mappage utilisateur FED_PROXY_USER, vous devez également spécifier l'option de serveur FED_PROXY_USER.</p> <p>Restriction : Cette option du serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS Version 9 et ultérieure.</p>
ACCOUNTING_STRING	<p>Obligatoire si les informations comptables doivent être transmises. Spécifie une chaîne comptable DRDA. Les valeurs valides incluent toute chaîne comprenant jusqu'à 255 caractères.</p>
REMOTE_AUTHID	<p>Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
REMOTE_PASSWORD	<p>Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
USE_TRUSTED_CONTEXT	<p>Indique si le mappage utilisateur est accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; le mappage utilisateur n'est pas accrédité et peut être uniquement utilisé dans des connexions sortantes fédérées non accréditées. Y indique que le mappage utilisateur est accrédité et peut être utilisé à la fois dans des connexions fédérées sortantes accréditées et non accréditées.</p> <p>Restriction : Cette option de serveur est uniquement valide pour DB2 Database pour Linux, UNIX et Windows version 9.5 et ultérieure et DB2 for z/OS version 9 et ultérieure.</p>

Options de colonne

Tableau 38. Options de colonne pour les sources de données DB2

Option	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
NO_EMPTY_STRING	Indique si le serveur source de données distant peut contenir des chaînes vides. Les valeurs valides sont Y et N. La valeur par défaut varie selon votre source de données distante. Pour les sources de données distantes Oracle, la valeur par défaut est Y ; toutes les valeurs de chaîne vides sont converties en valeurs NULL. Pour toutes les autres sources de données distantes, la valeur par défaut est N ; la source de données peut contenir des chaînes vides.
XML_ROOT	Spécifie l'élément racine XML à ajouter aux valeurs d'une colonne XML faisant référence à une séquence XML. Cette option veille à ce que les valeurs de la colonne XML forment un document XML syntaxiquement correct.

Référence des options Excel

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, et des options de pseudonyme.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 39. Options d'encapsuleur pour Excel

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Options de serveur

Tableau 40. Options de serveur pour Excel

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Options de pseudonyme

Tableau 41. Options de pseudonyme pour Excel

Option	Description
FILE_PATH	Obligatoire. Spécifie le nom de fichier et le chemin d'accès au répertoire qualifié complet du tableur Excel auquel vous souhaitez accéder.
RANGE	Spécifie la plage de cellules à utiliser, par exemple, A1:C100. La valeur avant les deux-points correspond à la cellule supérieure gauche de la plage de cellules. La valeur située après les deux-points correspond à la cellule inférieure droite de la plage de cellules.

Référence des options Informix

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 42. Options d'encapsuleur pour Informix

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 43. Options de serveur pour Informix

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 43. Options de serveur pour Informix (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DBNAME	Obligatoire. Spécifie le nom de la base de données Informix à laquelle vous souhaitez accéder.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données. Si plusieurs plans d'accès remplissent les critères, c'est le plan dont le coût est le moins élevé qui est choisi.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.

Tableau 43. Options de serveur pour Informix (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
INFORMIX_CLIENT_LOCALE	Spécifie la valeur CLIENT_LOCALE à utiliser pour la connexion entre le serveur fédéré et le serveur de source de données. La valeur correspond à tout environnement local Informix valide. Si vous ne spécifiez pas cette option, la variable d'environnement CLIENT_LOCALE est définie sur la valeur qui est spécifiée dans le fichier db2dj.ini. Si la variable db2dj.ini ne spécifie pas la variable d'environnement CLIENT_LOCALE, l'option INFORMIX_CLIENT_LOCALE est définie sur l'environnement local Informix qui correspond le plus à la page de codes et au territoire de la base de données fédérée.

Tableau 43. Options de serveur pour Informix (suite)

Nom	Description
INFORMIX_DB_LOCALE	Spécifie la variable d'environnement DB_LOCALE Informix à utiliser pour la connexion entre le serveur fédéré et le serveur de source de données. Si l'option INFORMIX_DB_LOCALE n'est pas spécifiée, la variable d'environnement DB_LOCALE Informix est définie sur la valeur spécifiée dans le fichier db2dj.ini. Si le fichier db2dj.ini ne spécifie aucune valeur, la variable d'environnement DB_LOCALEInformix DB_LOCALE n'est pas définie.
INFORMIX_LOCK_MODE	Spécifie le mode de verrouillage à définir pour une source de données Informix. L'encapsuleur Informix émet la commande SET LOCK MODE immédiatement après s'être connecté à une source de données Informix. Les valeurs valides sont W, N et un nombre. La valeur par défaut est W ; l'encapsuleur attend que le verrou soit relâché pendant une durée illimitée. N indique de ne pas attendre ; une erreur est immédiatement renvoyée. Utilisez un nombre pour spécifier la durée d'attente maximale, en secondes. Si un interblocage ou un délai d'attente se produit, utilisez l'instruction ALTER SERVER pour modifier la valeur de l'option INFORMIX_LOCK_MODE. Par exemple : ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.

Tableau 43. Options de serveur pour Informix (suite)

Nom	Description
IUD_APP_SVPT_ENFORCE	Indique si le serveur fédéré applique l'utilisation des instructions de point de sauvegarde des applications. Les valeurs valides sont Y et N. La valeur par défaut est Y ; si la source de données n'applique pas les instructions de point de sauvegarde des applications et qu'une erreur se produit lors d'une opération d'insertion, de mise à jour ou de suppression, le serveur fédéré annule la transaction et le code d'erreur SQL SQL1476N est renvoyé. Il est recommandé d'utiliser la valeur par défaut.
NODE	Obligatoire. Spécifie le nom selon lequel la source de données est définie en tant qu'instance dans son système de gestion de base de données relationnelle.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE=, les fonctions de colonne et fonctions scalaires, comme MAX et MIN, les tris, comme ORDER BY OR GROUP BY, et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.

Options de mappage utilisateur

Tableau 44. Options de mappage utilisateur pour Informix

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.

Tableau 44. Options de mappage utilisateur pour Informix (suite)

Nom	Description
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 45. Options de colonne pour Informix

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options JDBC

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 46. Options d'encapsuleur pour JDBC

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. La seule valeur valide est Y car le serveur DB2 ne prend en charge que le chargement de la machine virtuelle Java en mode protégé. La valeur par défaut est Y. L'encapsuleur est exécuté en mode protégé.

Tableau 46. Options d'encapsuleur pour JDBC (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 47. Options de serveur pour JDBC

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.

Tableau 47. Options de serveur pour JDBC (suite)

Nom	Description
DATEFORMAT	Spécifie le format de date que la source de données utilise. Utilisez 'DD', 'MM' et 'YY' ou 'YYYY' pour spécifier le format de date. Vous pouvez spécifier un délimiteur comme un espace, un trait d'union ou une virgule. Par exemple, le format 'YYYY-MM-DD' spécifie une date comme 1958-10-01. La valeur peut contenir des valeurs nulles.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 0. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 47. Options de serveur pour JDBC (suite)

Nom	Description
DRIVER_CLASS	<p>Indique la bibliothèque des pilotes JDBC. Vous pouvez enregistrer le serveur pour plusieurs sources de données JDBC si le pilote JDBC est compatible avec la version de spécification JDBC 3.0 ou une version ultérieure. Consultez la documentation de votre pilote JDBC pour obtenir des informations sur les spécifications JDBC et la configuration de l'option de serveur DRIVER_CLASS.</p> <p>Exemple</p> <p>Dans cet exemple, la bibliothèque de pilotes JDBC com.ibm.db2.jcc.DB2Driver est indiquée :</p> <pre>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</pre> <p>Important : Si vous définissez cette option, vous devez également définir l'option de serveur URL.</p>
DRIVER_PACKAGE	<p>Spécifie les modules des pilotes JDBC. Utilisez un séparateur de chemin d'accès pour spécifier plusieurs modules de classe de pilote. Utilisez un point-virgule sur les systèmes d'exploitation Windows et deux points sur les systèmes d'exploitation Linux et UNIX.</p> <p>Exemple</p> <p>Dans cet exemple, vous spécifiez plusieurs modules de pilotes avec deux points sur des systèmes d'exploitation Linux :</p> <pre>DRIVER_PACKAGE '/path1/file1.jar; /path2/file2.jar'</pre>
FOLD_ID	<p>Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.</p>
FOLD_PW	<p>Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.</p>

Tableau 47. Options de serveur pour JDBC (suite)

Nom	Description
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
JDBC_LOG	<p>Indique si l'encapsuleur JDBC crée des fichiers journaux pour la fonction de trace des erreurs. Les valeurs valides sont Y et N. La valeur par défaut est N. Il n'y pas de fichier journal créé. Si cette option de serveur est définie sur Y, l'encapsuleur JDBC enregistre les fichiers journaux JDBC dans le fichier <code>jdbc_wrapper_ID_produit.log</code> où <code>ID_produit</code> correspond à l'ID du produit. Le fichier journal est stocké dans le répertoire indiqué par le paramètre de configuration du gestionnaire de base de données DB2 DIAGPATH. Le répertoire par défaut sur les systèmes UNIX est <code>inst_home/sqllib/db2dump</code>.</p> <p>Recommandation : Définir cette option de serveur sur YES a un impact sur les performances de votre système. Il n'est pas conseillé d'activer la consignment sur des systèmes de production.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>

Tableau 47. Options de serveur pour JDBC (suite)

Nom	Description
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.
TIMEFORMAT	Spécifie le format horaire que la source de données utilise. Utilisez 'hh12', 'hh24', 'mm', 'ss', 'AM' et 'A.M.' pour spécifier le format horaire. Par exemple, le format 'hh24:mm:22' spécifie une heure telle que 16:00:00. Le format 'hh12:mm:ss AM' spécifie une heure telle que 8:00:00 AM. La valeur peut contenir des valeurs nulles.
TIMESTAMPFORMAT	Spécifie le format d'horodatage que la source de données utilise. Les valeurs valides sont au format utilisé par les options DATEFORMAT et TIMEFORMAT. Spécifiez 'n' pour un dixième de seconde, 'nn' pour un centième de seconde, 'nnn' pour des millisecondes, et ainsi de suite, jusqu'à 'nnnnnn' pour des microsecondes. Par exemple, le format 'YYY-MM-DD-hh24:mm:ss.nnnnnn' spécifie un horodatage de type 1994-01-01-24:00:00.000000. La valeur peut contenir des valeurs nulles.

Tableau 47. Options de serveur pour JDBC (suite)

Nom	Description
URL	<p>Spécifie la chaîne de connexion JDBC du serveur distant.</p> <p>La chaîne de connexion JDBC est constituée de trois parties, séparées par deux points :</p> <ul style="list-style-type: none"> • Protocole de base de données • Nom du type de base de données ou nom du driver de connectivité • Identité de la base de données, via un alias ou un sous-nom <p>Exemple</p> <p>Dans cet exemple, la chaîne de connexion JDBC est</p> <pre>jdbc:db2://cn.ibm.com:50471/testdb</pre> <p>:</p> <p>URL</p> <pre>'jdbc:db2://cn.ibm.com:50471/testdb'</pre> <p>Vous pouvez enregistrer le serveur pour plusieurs sources de données JDBC si le pilote JDBC est compatible avec la version de spécification JDBC 3.0 ou une version supérieure. Consultez la documentation de votre pilote JDBC pour obtenir des informations sur les spécifications JDBC et la configuration de l'option de serveur URL.</p> <p>Important : Si vous définissez cette option, vous devez également définir l'option de serveur DRIVER_CLASS.</p>
VARCHAR_NO_TRAILING_BLANKS	<p>Spécifie si la source de données contient des colonnes VARCHAR ayant au moins un caractère blanc de fin. La valeur par défaut est N ; les colonnes VARCHAR contiennent au moins un caractère blanc de fin.</p>

Options de mappage utilisateur

Tableau 48. Options de mappage utilisateur pour JDBC

Nom	Description
REMOTE_AUTHID	<p>Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>
REMOTE_PASSWORD	<p>Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.</p>

Options de colonne

Tableau 49. Options de colonne pour JDBC

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique s'il existe au moins un blanc de fin dans la colonne VARCHAR.

Référence des options Microsoft SQL Server

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 50. Options d'encapsuleur pour Microsoft SQL Server

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Tableau 50. Options d'encapsuleur pour Microsoft SQL Server (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 51. Options de serveur pour Microsoft SQL Server

Nom	Description
CODEPAGE	Spécifie la page de codes correspondant au jeu de caractères codés de la configuration client de la source de données. Sous les systèmes UNIX et Microsoft Windows qui utilisent une base de données fédérée non Unicode, la valeur par défaut est la page de codes de la base de données fédérée. Sous les systèmes UNIX qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1208. Sous les systèmes Windows qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1202.
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.

Tableau 51. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
COMM_RATE	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4.
DBNAME	Obligatoire. Spécifie les alias de la base de données à laquelle vous souhaitez accéder. La valeur est sensible à la casse.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données. Si plusieurs plans d'accès remplissent les critères, c'est le plan dont le coût est le moins élevé qui est choisi.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 51. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter. Important : Si vous définissez cette option sur Y, vous devez également spécifier XA_OPEN_STRING_OPTION.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 51. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NODE	Obligatoire. Si le serveur fédéré utilise Microsoft Windows, la valeur de NODE est le nom de système DSN que vous spécifiez pour Microsoft SQL Server. Si le serveur fédéré utilise UNIX ou Linux, la valeur de NODE est définie dans le fichier <code>odbc.ini</code> . La valeur est sensible à la casse.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.
PASSWORD	Spécifie si les mots de passe sont envoyés ou non à la source de données. La valeur par défaut est Y.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.

Tableau 51. Options de serveur pour Microsoft SQL Server (suite)

Nom	Description
XA_OPEN_STRING_OPTIONS	Obligatoire lorsque DB2_TWO_PHASE_COMMIT est défini sur Y. Spécifie l'ID du gestionnaire de ressources du registre Microsoft SQL Server.

Options de mappage utilisateur

Tableau 52. Options de mappage utilisateur pour Microsoft SQL Server

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 53. Options de colonne pour Microsoft SQL Server

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options ODBC

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 54. Options d'encapsuleur pour ODBC

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Important : Si vous définissez cette option sur Y pour un système UNIX, vous devez également définir l'option d'encapsuleur DB2_SOURCE_CLIENT_MODE.
DB2_SOURCE_CLIENT_MODE	Spécifie que le client de la source de données est en 32 bits et que l'instance de base de données sur le serveur fédéré est en 64 bits. La seule valeur valide est 32 bits. Cette option n'est valide que pour UNIX. Important : Si vous définissez cette option, vous devez également définir l'option d'encapsuleur DB2_FENCED sur Y.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
MODULE	Obligatoire pour les serveurs fédérés qui sont exécutés sur un système UNIX. Spécifie le chemin complet de la bibliothèque qui contient l'implémentation du gestionnaire de pilote ODBC ou l'implémentation SQL/CLI. Il n'existe pas de valeur par défaut pour UNIX. Sur un système Microsoft Windows, la valeur par défaut est odbc32.dll.

Options de serveur

Tableau 55. Options de serveur pour ODBC

Nom	Description
CODEPAGE	Spécifie la page de codes correspondant au jeu de caractères codés de la configuration client de la source de données. Sous les systèmes UNIX et Windows qui utilisent une base de données fédérée non Unicode, la valeur par défaut est la page de codes utilisée par la base de données fédérée. Sous les systèmes UNIX qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1208. Sous les systèmes Windows qui utilisent une base de données fédérée Unicode, la valeur par défaut est 1202.
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DATEFORMAT	Spécifie le format de date que la source de données utilise. Utilisez 'DD', 'MM' et 'YY' ou 'YYYY' pour spécifier le format de date. Vous pouvez spécifier un délimiteur comme un espace, un trait d'union ou une virgule. Par exemple, le format 'YYYY-MM-DD' spécifie une date comme 1958-10-01. La valeur peut contenir des valeurs nulles.

Tableau 55. Options de serveur pour ODBC (suite)

Nom	Description
DBNAME	Spécifie le nom de la base de données de la source de données à laquelle vous souhaitez accéder.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 0. -1 spécifie que l'optimiseur de requêtes fédéré détermine le nombre de demandes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 55. Options de serveur pour ODBC (suite)

Nom	Description
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NODE	Obligatoire. Spécifie le nom du noeud ou le nom DNS du système affecté à la source de données ODBC lorsque le DSN est défini. La valeur est sensible à la casse.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.

Tableau 55. Options de serveur pour ODBC (suite)

Nom	Description
TIMEFORMAT	Spécifie le format horaire que la source de données utilise. Utilisez 'hh12', 'hh24', 'mm', 'ss', 'AM' et 'A.M.' pour spécifier le format horaire. Par exemple, le format 'hh24:mm:22' spécifie une heure telle que 16:00:00. Le format 'hh12:mm:ss AM' spécifie une heure telle que 8:00:00 AM. La valeur peut contenir des valeurs nulles.
TIMESTAMPFORMAT	Spécifie le format d'horodatage que la source de données utilise. Les valeurs valides sont au format utilisé par les options DATEFORMAT et TIMEFORMAT. Spécifiez 'n' pour un dixième de seconde, 'nn' pour un centième de seconde, 'nnn' pour des millisecondes, et ainsi de suite, jusqu'à 'nnnnnn' pour des microsecondes. Par exemple, le format 'YYY-MM-DD-hh24:mm:ss.nnnnnn' spécifie un horodatage de type 1994-01-01-24:00:00.000000. La valeur peut contenir des valeurs nulles.
VARCHAR_NO_TRAILING_BLANKS	Spécifie si la source de données contient des colonnes VARCHAR ayant au moins un caractère blanc de fin. La valeur par défaut est N ; les colonnes VARCHAR contiennent au moins un caractère blanc de fin.

Options de mappage utilisateur

Tableau 56. Options de mappage utilisateur pour ODBC

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 57. Options de colonne pour ODBC

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique s'il existe au moins un blanc de fin dans la colonne VARCHAR.

Référence des options Oracle

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 58. Options d'encapsuleur pour Oracle

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Tableau 58. Options d'encapsuleur pour Oracle (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 59. Options de serveur pour Oracle

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.

Tableau 59. Options de serveur pour Oracle (suite)

Nom	Description
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. Important : L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD.

Tableau 59. Options de serveur pour Oracle (suite)

Nom	Description
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NODE	Obligatoire. Spécifie le nom du noeud où le serveur de base de données Oracle réside. Obtenez le nom du noeud à partir du fichier tnsnames.ora.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.

Tableau 59. Options de serveur pour Oracle (suite)

Nom	Description
PLAN_HINTS	Spécifie si des algorithmes d'optimisation de plan doivent être activés. Les algorithmes d'optimisation de plan sont des fragments d'instruction qui fournissent des informations supplémentaires que l'optimiseur de source de données utilise pour améliorer les performances de la requête. L'optimiseur de source de données utilise les algorithmes d'optimisation de plan pour décider s'il doit utiliser un index ou le type d'index ou de séquence de jointure de table à utiliser. Les valeurs valides sont Y et N. La valeur par défaut est N. Les algorithmes d'optimisation de plan doivent être activés. Y spécifie que les algorithmes d'optimisation de plan doivent être activés au niveau de la source de données s'ils sont pris en charge.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y. Le serveur fédéré autorise la source de données à évaluer les opérations. N spécifie que le serveur fédéré récupère les colonnes à partir de la source de données distante.
VARCHAR_NO_TRAILING_BLANKS	Indique pour un serveur particulier si des colonnes VARCHAR contiennent des blancs de fin. Pour appliquer l'option à une seule colonne, utilisez l'option de colonne VARCHAR_NO_TRAILING_BLANKS.
XA_OPEN_STRING_OPTIONS	Spécifie des informations supplémentaires à ajouter à la chaîne. Par exemple, il peut s'agir du répertoire pour les fichiers de trace.

Options de mappage utilisateur

Tableau 60. Options de mappage utilisateur pour Oracle

Nom	Description
FED_PROXY_USER	Spécifie l'ID d'autorisation à utiliser pour établir toutes les connexions accréditées sortantes lorsque la connexion entrante est non accréditée. Important : L'utilisateur dont l'ID est spécifié avec cette option doit avoir un mappage utilisateur spécifiant un REMOTE_AUTHID et un REMOTE_PASSWORD. Si vous spécifiez l'option de mappage utilisateur FED_PROXY_USER, vous devez également spécifier l'option de serveur FED_PROXY_USER.

Tableau 60. Options de mappage utilisateur pour Oracle (suite)

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.
USE_TRUSTED_CONTEXT	Indique si le mappage utilisateur est accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; le mappage utilisateur n'est pas accrédité et ne peut être utilisé que dans des connexions sortantes non accréditées. Y spécifie que le mappage utilisateur est accrédité et peut être utilisé dans les connexions sortantes accréditées et non accréditées.

Options de colonne

Tableau 61. Options de colonne pour Oracle

Nom	Description
NUMERIC_STRING	Spécifie si la colonne contient des chaînes de caractères numériques incluant des blancs. Les valeurs valides sont Y et N. La valeur par défaut est N ; la colonne ne contient pas de chaînes numériques incluant des blancs. Si la colonne contient uniquement des chaînes numériques suivies par des blancs de fin, ne spécifiez pas Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient pas de blancs susceptibles d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.
VARCHAR_NO_TRAILING_BLANKS	Indique si la colonne contient des blancs de fin.

Référence des options de script

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 62. Options d'encapsuleur pour les scripts

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.

Options de serveur

Tableau 63. Options de serveur pour les scripts

Nom	Description
DAEMON_PORT	Spécifie le numéro de port sur lequel le démon Script écoute les demandes de travaux Script. La valeur par défaut est 4099. Le numéro de port doit être le même que pour l'option DAEMON_PORT dans le fichier de configuration du démon. Si un nom de service est configuré pour le démon Script, utilisez un nom de service TCP/IP.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 63. Options de serveur pour les scripts (suite)

Nom	Description
NODE	Obligatoire. Spécifie le nom d'hôte DNS ou l'adresse IP du système sur lequel le démon Script est exécuté. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE et SOCKS. La valeur par défaut est NONE.

Options de mappage utilisateur

Tableau 64. Options de mappage utilisateur pour les scripts

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.

Options de pseudonyme

Tableau 65. Options de pseudonyme pour les scripts

Nom	Description
DATASOURCE	Obligatoire pour le pseudonyme racine. Spécifie le nom du script à appeler. Le script que vous spécifiez comme valeur de cette option doit également être spécifié dans le fichier de configuration du démon Script. Important : Cette option n'est valide que pour le pseudonyme racine.

Tableau 65. Options de pseudonyme pour les scripts (suite)

Nom	Description
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.</p>
TIMEOUT	<p>Spécifie la durée d'attente maximale, en minutes, des résultats provenant du démon. La valeur par défaut est 60. Cette option n'est valide que pour le pseudonyme racine.</p>
VALIDATE	<p>Spécifie si le document source est validé pour garantir sa conformité au schéma XML ou à une définition de type de document (DTD) avant d'en extraire des données. La valeur par défaut est N ; la validation n'a pas lieu. Avant de définir la valeur sur Y, le fichier schéma ou le fichier DTD se trouve à l'emplacement spécifié par le document source. Cette option n'est valide que pour le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.</p>
XPATH	<p>Spécifie l'expression XPath qui identifie les éléments XML représentant des tuples individuels. L'option de pseudonyme XPATH pour un pseudonyme enfant est évaluée dans le contexte du chemin spécifié par l'option de pseudonyme XPATH de son parent. Cette expression XPath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par la présence de l'option de colonne XPATH.</p>

Options de colonne

Tableau 66. Options de colonne pour les scripts

Nom	Description
DEFAULT	Spécifie une valeur par défaut d'une colonne d'entrée de script. Si la requête SQL ne fournit pas de valeur, la valeur par défaut est utilisée.
FOREIGN_KEY	Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne peut être utilisée pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY. Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.
INPUT_MODE	Indique le mode entrée de la colonne. Les valeurs valides sont CONFIG et FILE_INPUT. CONFIG traite la valeur comme le mode entrée d'une colonne. FILE_INPUT spécifie un fichier qui stocke la valeur. L'encapsuleur transmet la valeur spécifiée au démon Script.
POSITION	Spécifie un entier pour les paramètres positionnels. Si cette valeur positionnelle est définie sur un entier, alors cette entrée doit être dans cette position dans la ligne de commande. Si cette option est définie, le commutateur est inséré à l'emplacement approprié lorsque la requête est exécutée. Si POSITION est défini sur -1, l'option est ajoutée comme la dernière option de ligne de commande. Un entier POSITION ne peut pas être utilisé deux fois dans le même pseudonyme. Cette option ne s'applique qu'aux colonnes d'entrée.

Tableau 66. Options de colonne pour les scripts (suite)

Nom	Description
PRIMARY_KEY	Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant. Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.
SWITCH	Spécifie un indicateur pour le script sur la ligne de commande. La valeur de cette option précédait la valeur de la colonne fournie par WSSCRIPT.ARGS ou par la valeur par défaut, le cas échéant. Si vous ne spécifiez pas de valeur pour cette option et qu'une valeur par défaut existe pour la colonne, la valeur par défaut est ajoutée sans information de commutation. Cette option est obligatoire pour les colonnes d'entrée.
SWITCH_ONLY	Permet l'utilisation de commutateurs sans argument de ligne de commande. Les valeurs valides sont Y et N. Définissez cette option sur Y et les valeurs d'entrée valides sont Y et N. Pour une valeur d'entrée Y, seul le commutateur est ajouté à la ligne de commande. Pour une valeur d'entrée N, aucune valeur n'est ajoutée à la ligne de commande.
VALID_VALUES	Spécifie un ensemble de valeurs valides pour une colonne. Utilisez un point-virgule pour séparer plusieurs valeurs.
XPATH	Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.

Référence des options Sybase

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 67. Options d'encapsuleur pour Sybase

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Sous Microsoft Windows, les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Sous UNIX, la valeur par défaut et la seule valeur valide est Y ; l'encapsuleur doit fonctionner en mode protégé.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 68. Options de serveur pour Sybase

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 68. Options de serveur pour Sybase (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
CONV_EMPTY_STRING	Spécifie si le serveur fédéré convertit une chaîne vide en un espace lors de tâches de réplication. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré ne convertit pas les chaînes vides. Définissez cette option sur Y lorsque la source de données présente une colonne avec un caractère ne pouvant avoir la valeur NULL qui stocke une chaîne vide.
DBNAME	Obligatoire. Spécifie le nom de la base de données à laquelle vous souhaitez accéder. Obtenez le nom de la base de données à partir du serveur Sybase.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 68. Options de serveur pour Sybase (suite)

Nom	Description
DB2_TWO_PHASE_COMMIT	Spécifie si le serveur fédéré se connecte à la source de données dans un protocole de validation à deux phases ou un protocole de validation à une phase. Les valeurs valides sont Y et N. La valeur par défaut est N ; le serveur fédéré utilise le protocole de validation à une phase pour se connecter. Y spécifie que le serveur fédéré utilise un protocole de validation à deux phases pour se connecter.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
FOLD_ID	Spécifie la casse de l'ID utilisateur envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie l'ID utilisateur en majuscule ; si l'ID utilisateur en majuscule échoue, le serveur envoie alors l'ID utilisateur en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.
FOLD_PW	Spécifie la casse du mot de passe envoyé à la source de données. Il n'y a pas de valeur par défaut ; le serveur fédéré envoie le mot de passe en majuscule ; si le mot de passe en majuscule échoue, le serveur envoie alors le mot de passe en minuscule. Les valeurs valides sont U (majuscule), L (minuscule) et N (null). Evitez d'utiliser le paramètre null, qui peut avoir un impact négatif sur les performances.

Tableau 68. Options de serveur pour Sybase (suite)

Nom	Description
IFILE	<p>Spécifie le chemin d'accès et le nom du fichier d'interface Sybase à utiliser à la place du fichier d'interface par défaut.</p> <p>L'encapsuleur Sybase recherche le fichier d'interface dans les emplacements suivants, dans l'ordre spécifié : sous Microsoft Windows, dans l'option de serveur IFILE, puis dans le répertoire %DB2PATH%\interfaces et enfin dans le répertoire %SYBASE%\ini\sql.ini. Sous UNIX, dans l'option de serveur IFILE, puis dans le répertoire sqllib/interfaces et enfin dans le répertoire \$SYBASE/interfaces.</p>
IO_RATIO	<p>Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23}. La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.</p>
LOGIN_TIMEOUT	<p>Spécifie la durée d'attente, en secondes, du serveur fédéré avant d'abandonner une demande de connexion. La valeur par défaut est 0 ; le serveur fédéré attend pendant une durée illimitée.</p>
NODE	<p>Obligatoire. Spécifie le nom du noeud où le serveur Sybase réside. Le nom de noeud est dans le fichier des interfaces Sybase.</p>
OLD_NAME_GEN	<p>Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.</p>

Tableau 68. Options de serveur pour Sybase (suite)

Nom	Description
PACKET_SIZE	Spécifie la taille de paquet, en octets, que la bibliothèque client utilise pour envoyer des paquets TDS (Tabular Data Stream). Si l'encapsuleur Sybase a besoin d'envoyer ou de recevoir de grandes quantités de données image et texte, augmentez la valeur de PACKET_SIZE.
PLAN_HINTS	Spécifie si des algorithmes d'optimisation de plan doivent être activés. Les algorithmes d'optimisation de plan sont des fragments d'instruction qui fournissent des informations supplémentaires que l'optimiseur de source de données utilise pour améliorer les performances de la requête. L'optimiseur de source de données utilise les algorithmes d'optimisation de plan pour décider s'il doit utiliser un index ou le type d'index ou de séquence de jointure de table à utiliser. Les valeurs valides sont Y et N. La valeur par défaut est N. Les algorithmes d'optimisation de plan doivent être activés. Y spécifie que les algorithmes d'optimisation de plan doivent être activés au niveau de la source de données s'ils sont pris en charge.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.
TIMEOUT	Spécifie la durée maximale, en secondes, au cours de laquelle le serveur fédéré attend que le serveur distant réponde à une commande. La valeur par défaut est 0, correspondant à une durée illimitée.
XA_OPEN_STRING_OPTIONS	Spécifie des chaînes ouvertes pour l'interface Sybase DTM XA. Ces chaînes s'ajoutent au nom LRM, au nom d'utilisateur et au mot de passe.

Options de mappage utilisateur

Tableau 69. Options de mappage utilisateur pour Sybase

Option	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 70. Options de colonne pour Sybase

Option	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options Teradata

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 71. Options d'encapsuleur pour Teradata

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Sous Microsoft Windows, les valeurs valides sont Y et N. La valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité. Sous UNIX, la valeur par défaut Y ; l'encapsuleur doit fonctionner en mode protégé.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Options de serveur

Tableau 72. Options de serveur pour Teradata

Nom	Description
COLLATING_SEQUENCE	Spécifie si la source de données utilise la même séquence de classement par défaut que la base de données fédérée. Les valeurs valides sont Y, N et I. I spécifie une insensibilité à la casse. La valeur par défaut est Y. La séquence de classement spécifiée pour le serveur fédéré doit correspondre à la séquence de classement sur la source de données distante.
COMM_RATE	Spécifie le débit de communication, en mégaoctets par seconde, entre le serveur fédéré et le serveur source de données. Les valeurs valides sont supérieures à 0 et inférieures à 2147483648. La valeur par défaut est 2.

Tableau 72. Options de serveur pour Teradata (suite)

Nom	Description
CPU_RATIO	Spécifie combien l'unité centrale de la source de données est plus rapide ou plus lente que l'unité centrale du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'unité centrale ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse de l'unité centrale du serveur fédéré est 50 % inférieure à celle de l'unité centrale de la source de données. Un paramétrage de 2 indique que l'unité centrale du serveur fédéré est deux fois plus rapide que l'unité centrale de la source de données.
DB2_MAXIMAL_PUSHDOWN	Spécifie les critères principaux que l'optimiseur de requêtes utilise pour choisir un plan d'accès. Les valeurs valides sont Y et N. La valeur par défaut est N ; l'optimiseur de requêtes choisit le plan ayant le coût estimé le plus faible. Y spécifie que l'optimiseur de requêtes choisit le plan d'accès qui transfère l'exécution des opérations de requête vers la source de données.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.

Tableau 72. Options de serveur pour Teradata (suite)

Nom	Description
IO_RATIO	Spécifie combien la vitesse d'exécution du système d'E/S de la source de données est plus élevée ou plus faible que celle du système d'E/S du serveur fédéré. Les valeurs valides sont supérieures à 0 et inférieures à 1×10^{23} . La valeur par défaut est 1.0. Les valeurs peuvent être exprimées dans n'importe quelle double notation valide, par exemple, 123E10, 123 ou 1.21E4. Un paramétrage de 1 indique que le serveur fédéré et le serveur de source de données ont la même vitesse d'E/S ; un ratio de 1:1. Un paramétrage de 0,5 indique que la vitesse du serveur fédéré est 50 % inférieure à celle de la source de données. Un paramétrage de 2 indique que la vitesse fédérée est deux fois plus élevée que la vitesse de la source de données.
NODE	Obligatoire. Spécifie le nom d'alias ou l'adresse IP du serveur Teradata.
OLD_NAME_GEN	Spécifie comment convertir les noms de colonne et d'index de la source de données en noms de colonne de pseudonyme et noms d'index local du serveur fédéré. Les valeurs valides sont Y et N. La valeur par défaut est N ; les noms générés correspondent étroitement aux noms de la source de données. Y spécifie que les noms générés sont les mêmes que les noms créés dans IBM WebSphere Federation Server Version 9 et versions antérieures. Ainsi, il est possible que les noms ne correspondent pas étroitement aux noms de la source de données.
PUSHDOWN	Spécifie si le serveur fédéré autorise la source de données à évaluer les opérations. Les valeurs valides sont Y et N. La valeur par défaut est Y ; la source de données évalue les opérations. N spécifie que le serveur fédéré envoie les instructions SQL qui ne contiennent que SELECT avec les noms de colonne. Les prédicats, comme WHERE= ; les fonctions de colonne et fonctions scalaires, comme MAX et MIN ; les tris, comme ORDER BY OR GROUP BY ; et les jointures ne sont pas inclus dans les instructions SQL que le serveur fédéré envoie à la source de données.

Options de mappage utilisateur

Tableau 73. Options de mappage utilisateur pour Teradata

Nom	Description
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.

Options de colonne

Tableau 74. Options de colonne pour Teradata

Nom	Description
NUMERIC_STRING	Indique comment gérer les chaînes numériques. La valeur par défaut est N. Si la colonne de chaîne de source de données ne contient que des chaînes numériques et aucun autre caractère, y compris les espaces vides, définissez l'option NUMERIC_STRING sur Y. Lorsque NUMERIC_STRING est défini sur Y pour une colonne, l'optimiseur de requêtes reconnaît que la colonne ne contient aucun espace vide susceptible d'interférer dans le tri des données de la colonne. Utilisez cette option lorsque la séquence de classement d'une source de données est différente de la séquence de classement utilisée par le serveur fédéré. Les colonnes qui utilisent cette option ne sont pas exclues de l'évaluation distante en raison d'une séquence de classement différente.

Référence des options des fichiers structurés sous forme de tables

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un pseudonyme et des options de colonne.

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Options d'encapsuleur

Tableau 75. Options d'encapsuleur pour les fichiers structurés sous forme de tables

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.

Options de serveur

Tableau 76. Options de serveur pour les fichiers structurés sous forme de tables

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Options de pseudonyme

Tableau 77. Options de pseudonyme pour les fichiers structurés sous forme de tables

Nom	Description
COLUMN_DELIMITER	Spécifie un caractère unique à utiliser comme délimiteur pour séparer les colonnes dans le fichier structuré sous forme de tables. La valeur par défaut est une virgule (.). Un guillemet simple ne peut pas être utilisé comme délimiteur. Le délimiteur de colonne doit être cohérent sur l'ensemble du fichier. Une valeur NULL est représentée par deux délimiteurs l'un à côté de l'autre ou par un délimiteur suivi d'un caractère de fin de ligne, si la zone NULL est la dernière de la ligne. Le délimiteur de colonne ne peut pas exister en tant que donnée valide d'une colonne.
CODEPAGE	Spécifie la page de codes du fichier au niveau de la source de données. Cette option n'est valide que pour les bases de données fédérées qui utilisent Unicode. Les données source sont converties à partir de la page de codes spécifiée en Unicode.

Tableau 77. Options de pseudonyme pour les fichiers structurés sous forme de tables (suite)

Nom	Description
FILE_PATH	<p>Indique le chemin qualifié complet vers le fichier structuré sous forme de table. Encadrez le nom de fichier par des guillemets simples. Le fichier de données doit être un fichier standard ou un lien symbolique, plutôt qu'un canal ou un autre type de fichier non standard.</p> <p>Important : Si vous spécifiez l'option FILE_PATH, ne spécifiez pas une colonne DOCUMENT.</p>
KEY_COLUMN	<p>Spécifie le nom de la colonne sur laquelle le fichier est stocké. Une colonne ayant l'option de colonne DOCUMENT ne peut pas être la colonne clé. Seules les clés à colonne unique sont prises en charge. La valeur doit être le nom d'une colonne définie dans l'instruction CREATE NICKNAME. La colonne doit être triée dans l'ordre croissant. La colonne clé doit être désignée comme ne pouvant avoir la valeur NULL en ajoutant l'option NOT NULL à ses définitions dans l'instruction de pseudonyme. Cette valeur fait la distinction majuscules/minuscules.</p>
SORTED	<p>Spécifie si le fichier au niveau de la source de données est ou n'est pas trié dans l'ordre croissant. Les valeurs valides sont Y et N. La valeur par défaut est N. Le fichier au niveau de la source de données n'est pas trié dans l'ordre croissant. Les sources de données triées doivent être triées dans l'ordre croissant conformément à la séquence de classement de l'environnement local en cours, comme défini par les paramètres de la catégorie Prise en charge de la langue nationale LC_COLLATE. Si vous spécifiez que la source de données est triée, définissez l'option VALIDATE_DATA_FILE sur Y.</p>
VALIDATE_DATA_FILE	<p>Pour les fichiers triés, cette option indique si l'encapsuleur vérifie que la colonne de clé est triée dans l'ordre croissant et contrôle la présence de clés ayant la valeur NULL. Cette validation n'a lieu qu'une seule fois lorsque le pseudonyme est créé pour la première fois. La valeur par défaut est N ; l'ordre de tri n'est pas vérifié. Cette option n'est valide que lorsque l'option SORTED est définie sur Y et que l'option DOCUMENT n'est pas spécifiée.</p>

Options de colonne

Tableau 78. Options de colonne pour les fichiers structurés sous forme de tables

Option	Description
DOCUMENT	Vous permet de spécifier le chemin du fichier lors de l'exécution de la requête, plutôt que lorsque vous créez le pseudonyme. La seule valeur valide est FILE. Seule une colonne de chaque pseudonyme peut être spécifiée avec l'option DOCUMENT. La colonne associée à l'option DOCUMENT doit être un type de données VARCHAR ou CHAR. L'utilisation d'une option de colonne de pseudonyme DOCUMENT plutôt qu'une option de pseudonyme FILE_PATH implique que le fichier qui correspond à ce pseudonyme sera fourni lors de l'exécution de la requête. Si l'option DOCUMENT a la valeur FILE, la valeur fournie lors de l'exécution de la requête est le chemin complet du fichier dont le schéma correspond à la définition de pseudonyme pour ce pseudonyme.

Référence des options des services Web

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 79. Options d'encapsuleur pour les services Web

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.

Tableau 79. Options d'encapsuleur pour les services Web (suite)

Nom	Description
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé. La valeur par défaut est <i>install path/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <i>file:file_name</i> , où <i>file_name</i> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de serveur

Tableau 80. Options de serveur pour les services Web

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.

Tableau 80. Options de serveur pour les services Web (suite)

Nom	Description
DB2_UM_PLUGIN	Spécifie l'implémentation du plug-in de mappage utilisateur. Pour un plug-in écrit en Java, cette option spécifie une chaîne sensible à la casse pour le nom de classe correspondant à la classe du référentiel de mappage utilisateur. Par exemple, "UserMappingRepositoryLDAP". Pour un plug-in écrit en langage C, elle spécifie n'importe quel nom de bibliothèque C valide.
DB2_UM_PLUGIN_LANG	Spécifie la langue du plug-in de mappage utilisateur. Les valeurs valides sont Java et C. La valeur par défaut est Java.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé. La valeur par défaut est <i>install path/cfg/WSWrapperKeystore.kdb</i> .

Tableau 80. Options de serveur pour les services Web (suite)

Nom	Description
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <code>file:file_name</code> , où <code>file_name</code> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de mappage utilisateur

Tableau 81. Options de mappage utilisateur pour les services Web

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy. Le mot de passe est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée.
REMOTE_AUTHID	Spécifie l'ID utilisateur distant vers lequel l'ID utilisateur local est mappé. Si vous ne spécifiez pas cette option, l'ID permettant d'établir une connexion vers la base de données fédérée est utilisé.
REMOTE_PASSWORD	Spécifie le mot de passe distant de l'ID utilisateur distant. Si vous ne spécifiez pas cette option, le mot de passe permettant d'établir une connexion vers la base de données fédérée est utilisé.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.

Options de pseudonyme

Tableau 82. Options de pseudonyme pour les services Web

Nom	Description
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name'</pre>
SOAPACTION	<p>Obligatoire pour le pseudonyme racine. Spécifie l'attribut URI SOAPACTION à partir du format WSDL (Web Services Description Language). L'URL peut contenir une adresse IPv6 séparée par deux-points si elle est encadrée par des crochets. Par exemple :<code>http://[1080:0:0:0:8:800:200C:417A]</code></p> <p>Remarque : Cette option n'est pas valide pour les pseudonymes non racine.</p>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine.</p>
TEMPLATE	<p>Spécifie le fragment de modèle du pseudonyme à utiliser pour établir une requête SOAP. Le fragment doit être conforme à la syntaxe du modèle spécifié. Cette option n'est valide que pour le pseudonyme racine.</p>
URL	<p>Obligatoire pour le pseudonyme racine. Spécifie l'URL du noeud final du service Web. Les protocoles pris en charge sont HTTP et HTTPS. L'URL peut contenir une adresse IPv6 séparée par deux-points si elle est encadrée par des crochets. Par exemple :<code>http://[1080:0:0:0:8:800:200C:417A]</code></p>

Tableau 82. Options de pseudonyme pour les services Web (suite)

Nom	Description
XML_CODESET	Spécifie le codage à utiliser pour envoyer et recevoir des données XML. Cette option écrase le codage interne.
XPATH	Obligatoire. Spécifie l'expression Xpath qui identifie les éléments de réponse SOAP représentant des tuples individuels. Cette expression Xpath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par les options de colonne de pseudonyme XPATH.

Options de colonne

Tableau 83. Options de colonne pour les services Web

Nom	Description
ESCAPE_INPUT	Spécifie si les caractères spéciaux XML sont remplacés dans les valeurs d'entrée XML ou non. Utilisez cette option pour inclure des fragments XML comme entrée, par exemple, des fragments XML avec des éléments de répétition. Les valeurs valides sont Y et N. La valeur par défaut est N ; les valeurs d'entrée XML sont conservées. Le type de données de colonne doit être VARCHAR ou CHAR. Si ESCAPE_INPUT est défini sur Y, vous devez également spécifier l'option de colonne TEMPLATE.

Tableau 83. Options de colonne pour les services Web (suite)

Nom	Description
FOREIGN_KEY	<p>Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.
PRIMARY_KEY	<p>Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.

Tableau 83. Options de colonne pour les services Web (suite)

Nom	Description
SOAPACTIONCOLUMN	<p>Indique la colonne qui spécifie de manière dynamique l'action SOAP pour le noeud final des services Web lors de l'exécution d'une requête. Cette option n'est valide que pour le pseudonyme racine. Si le nom d'hôte est une adresse IPv6 (séparée par deux-points), encadrez le nom d'hôte par des crochets. Par exemple : 'http://[1080:0:0:0:8:800:200C:417A]:99/soap' Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.</p>
TEMPLATE	<p>Spécifie le fragment de modèle de colonne à utiliser pour établir un document d'entrée XML. Le fragment doit être conforme à la syntaxe du modèle spécifié.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>
URLCOLUMN	<p>Indique la colonne qui spécifie de manière dynamique l'action SOAP pour le noeud final des services Web lors de l'exécution d'une requête. Cette option n'est valide que pour le pseudonyme racine. Si le nom d'hôte est une adresse IPv6 (séparée par deux-points), encadrez le nom d'hôte par des crochets. Par exemple : 'http://[1080:0:0:0:8:800:200C:417A]:99/soap' Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne.</p>
XPATH	<p>Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>

Référence des options XML

Ces options permettent de configurer la manière dont le serveur fédéré et ses utilisateurs interagissent avec une source de données, de définir et modifier un encapsuleur, un serveur, un mappage utilisateur, un pseudonyme et des options de colonne.

Options d'encapsuleur

Les tableaux suivants répertorient les options qui s'appliquent à cette source de données et identifient les options requises que vous devez spécifier.

Tableau 84. Options d'encapsuleur pour XML

Nom	Description
DB2_FENCED	Obligatoire. Spécifie si l'encapsuleur fonctionne en mode protégé ou en mode accrédité. Les valeurs valides sont Y et N. la valeur par défaut est N ; l'encapsuleur fonctionne en mode accrédité.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé.
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et <code>file:file_name</code> , où <code>file_name</code> est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. Les valeurs valides sont Y et N. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de serveur

Tableau 85. Options de serveur pour XML

Nom	Description
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Spécifie le nombre maximum de demandes asynchrones simultanées à partir d'une requête. Les valeurs valides vont de -1 à 64000. La valeur par défaut est 1. -1 indique si l'optimiseur de requêtes fédéré détermine le nombre de requêtes. 0 spécifie que la source de données ne peut pas héberger des demandes asynchrones supplémentaires.
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
PROXY_SERVER_NAME	Spécifie le nom ou l'adresse IP du serveur proxy. Les adresses IP valides sont au format IPv4 (séparées par un point) ou au format IPv6 (séparées par deux-points). Utilisez le format IPv6 uniquement si IPv6 est configuré.
PROXY_SERVER_PORT	Spécifie le port ou le nom de service du service proxy sur le serveur proxy. Les valeurs valides sont un numéro de port décimal allant de 1 à 32760 ou un nom de service.
PROXY_TYPE	Spécifie le type de proxy à utiliser pour accéder à Internet lorsque le serveur fédéré se trouve derrière le pare-feu. Les valeurs valides sont NONE, HTTP et SOCKS. La valeur par défaut est NONE.
SOCKET_TIMEOUT	Spécifie la durée maximale, en minutes, au cours de laquelle le serveur fédéré attend les résultats provenant du serveur proxy. Les valeurs valides sont des nombres supérieurs ou égaux à 0. La valeur par défaut est 0 ; le serveur attend pendant une durée illimitée.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.
SSL_KEYSTORE_FILE	Spécifie le fichier de stockage pour les communications qui utilisent SSL ou TSL. Une valeur valide est un nom de chemin qualifié complet accessible par l'agent de base de données fédérée ou par un processus en mode protégé.

Tableau 85. Options de serveur pour XML (suite)

Nom	Description
SSL_KEYSTORE_PASSWORD	Spécifie le mot de passe à utiliser pour accéder au fichier dans l'option SSL_KEYSTORE_FILE. Les valeurs valides sont un mot de passe, qui est chiffré lorsqu'il est stocké dans le catalogue de la base de données fédérée, et file:file_name, où file_name est le chemin qualifié complet vers un fichier de dissimulation.
SSL_VERIFY_SERVER_CERTIFICATE	Spécifie si le certificat du serveur est vérifié lors de l'authentification SSL. La valeur par défaut est N ; le certificat n'est pas vérifié.

Options de mappage utilisateur

Tableau 86. Options de mappage utilisateur pour XML

Nom	Description
PROXY_AUTHID	Spécifie le nom d'utilisateur pour l'authentification du serveur proxy.
PROXY_PASSWORD	Spécifie le mot de passe pour l'authentification du serveur proxy.
SSL_CLIENT_CERTIFICATE_LABEL	Spécifie le certificat client à envoyer lors de l'authentification SSL. Si vous ne spécifiez pas de valeur, l'ID d'autorisation en cours de la base de données fédérée est utilisé pour déterminer l'emplacement du certificat.

Options de pseudonyme

Tableau 87. Options de pseudonyme pour XML

Nom	Description
DIRECTORY_PATH	Spécifie le nom de chemin d'un répertoire contenant un ou plusieurs fichiers XML. Utilisez cette option pour créer un pseudonyme unique sur de multiples fichiers source XML. L'encapsuleur XML n'utilise que des fichiers avec une extension .xml se trouvant dans le répertoire spécifié. L'encapsuleur XML ignore tous les autres fichiers du répertoire. Si vous spécifiez cette option de pseudonyme, ne spécifiez pas une colonne DOCUMENT. Cette option n'est valide que pour le pseudonyme racine.
FILE_PATH	Spécifie le chemin de fichier du document XML. Si vous spécifiez FILE_PATH, ne spécifiez pas une colonne DOCUMENT. Cette option n'est valide que pour le pseudonyme racine.

Tableau 87. Options de pseudonyme pour XML (suite)

Nom	Description
INSTANCE_PARSE_TIME	<p>Spécifie la durée nécessaire, en millisecondes, pour procéder à l'analyse syntaxique d'une ligne du document source XML. La valeur valide peut être un entier ou une valeur décimale. La valeur par défaut est 7. Cette option n'est valide que pour les colonnes du pseudonyme racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.</p>
NAMESPACES	<p>Spécifie les espaces de nom associés aux préfixes d'espace de nom utilisés dans les options XPATH et TEMPLATE pour chaque colonne. Utilisez cette syntaxe :</p> <pre>NAMESPACES'prefix1="actual_namespace1", prefix2="actual_namespace2"'</pre> <p>Utilisez une virgule pour séparer plusieurs espaces de nom. Par exemple :</p> <pre>NAMESPACES='http://www.myweb.com/cust', i='http://www.myweb.com/cust/id', n='http://www.myweb.com/cust/name''</pre>
NEXT_TIME	<p>Spécifie la durée nécessaire, en millisecondes, pour déterminer l'emplacement des éléments source consécutifs à partir de l'expression XPath. La valeur par défaut est 1. Cette option est valide pour les pseudonymes racine et non racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.</p>
STREAMING	<p>Spécifie si le document source doit être séparé en fragments logiques pour le traitement. Les fragments concordent avec le noeud correspondant à l'expression XPath du pseudonyme. L'encapsuleur procède ensuite à une analyse syntaxique et traite les données source fragment par fragment. Ce type d'analyse syntaxique minimise l'utilisation de la mémoire. Les valeurs valides sont Y et N. La valeur par défaut est N ; les documents ne sont pas soumis à une analyse syntaxique. Cette option n'est valide que sur le pseudonyme racine.</p>

Tableau 87. Options de pseudonyme pour XML (suite)

Nom	Description
VALIDATE	Spécifie si le document source est validé pour garantir sa conformité au schéma XML ou à une définition de type de document (DTD) avant d'en extraire des données. La valeur par défaut est N ; la validation n'a pas lieu. Avant de définir la valeur sur Y, le fichier schéma ou le fichier DTD se trouve à l'emplacement spécifié par le document source. Cette option n'est valide que pour le pseudonyme racine. Ne définissez pas les deux options STREAMING et VALIDATE sur Y.
XPATH	Obligatoire. Spécifie l'expression Xpath qui identifie les éléments représentant des tuples individuels. Cette expression Xpath est utilisée comme un contexte permettant d'évaluer les valeurs de colonne identifiées par l'option de colonne XPATH.
XPATH_EVAL_TIME	Spécifie la durée nécessaire, en millisecondes, pour évaluer l'expression XPath du pseudonyme et déterminer l'emplacement du premier élément. La valeur peut être un entier ou une valeur décimale. La valeur par défaut est 1. Cette option est valide pour les pseudonymes racine et non racine. Pour optimiser les requêtes de structures de source XML complexes ou volumineuses, modifiez les options INSTANCE_PARSE_TIME, XPATH_EVAL_TIME et NEXT_TIME.

Options de colonne

Tableau 88. Options de colonne pour XML

Nom	Description
DOCUMENT	Indique que cette colonne est une colonne DOCUMENT. La valeur de la colonne DOCUMENT indique le type de données source XML fourni au pseudonyme lors de l'exécution de la requête. Cette option n'est valide que pour les colonnes du pseudonyme racine (le pseudonyme qui identifie les éléments au niveau supérieur du document XML). Seule une colonne pour chaque pseudonyme peut être spécifiée avec l'option DOCUMENT. Si vous utilisez une option de colonne DOCUMENT plutôt qu'une option de pseudonyme FILE_PATH ou DIRECTORY_PATH, le document qui correspond à ce pseudonyme sera fourni lors de l'exécution de la requête. Les valeurs valides sont FILE, DIRECTORY, URI et COLUMN. La valeur par défaut est FILE. La colonne doit être un type de données VARCHAR.

Tableau 88. Options de colonne pour XML (suite)

Nom	Description
FOREIGN_KEY	<p>Indique que ce pseudonyme est un pseudonyme enfant et spécifie le nom du pseudonyme parent correspondant. Un pseudonyme peut avoir au maximum une option de colonne FOREIGN_KEY. La valeur de l'option est sensible à la casse. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer un pseudonyme parent et un pseudonyme enfant. Une instruction CREATE NICKNAME qui comprend une option FOREIGN_KEY échoue si le pseudonyme parent a un nom de schéma différent. A moins que le pseudonyme auquel il est fait référence dans une clause FOREIGN_KEY ait été explicitement défini en minuscule ou dans une casse mixte dans l'instruction CREATE NICKNAME, vous devez spécifier le pseudonyme en majuscule lorsque vous faites référence à ce pseudonyme dans la clause FOREIGN_KEY.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.
PRIMARY_KEY	<p>Obligatoire pour un pseudonyme parent ayant un ou plusieurs pseudonymes enfant. Spécifie que ce pseudonyme est un pseudonyme parent. Le type de données de colonne doit être VARCHAR(16). Un pseudonyme ne peut avoir qu'une seule option de colonne PRIMARY_KEY. Yes est la seule valeur valide. Ne spécifiez pas l'option XPATH pour cette colonne. La colonne ne peut être utilisée que pour associer des pseudonymes parent et des pseudonymes enfant.</p> <p>Remarque :</p> <ul style="list-style-type: none"> • Lorsque cette option est définie sur une colonne, aucune autre option ne peut être définie sur la colonne. • Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.

Tableau 88. Options de colonne pour XML (suite)

Nom	Description
XPATH	<p>Spécifie l'expression Xpath du document XML qui contient les données correspondant à cette colonne. L'encapsuleur évalue cette expression XPath après l'application par l'instruction CREATE NICKNAME de l'expression XPath à partir de l'option de pseudonyme XPATH.</p> <p>Remarque : Si vous utilisez cette option de colonne, vous ne pourrez pas utiliser ultérieurement l'instruction ALTER NICKNAME pour supprimer l'option. En effet, vous devrez supprimer le pseudonyme puis le créer à nouveau sans cette option de colonne.</p>

Chapitre 37. Vues de la table du catalogue global contenant des informations fédérées

La plupart des vues de catalogue dans une base de données fédérée sont identiques à celles de toute autre base de données DB2 for Linux, UNIX et Windows.

Il existe plusieurs vues uniques contenant des informations importantes pour un système fédéré, telle que la vue SYSCAT.WRAPPERS.

Les vues SYSCAT sont en lecture seule. Vous ne pouvez pas effectuer une opération de mise à jour ou de suppression sur une vue du système SYSCAT. Il est recommandé d'utiliser les vues SYSSTAT pour mettre le catalogue système à jour. Modifiez les applications qui font référence à la vue SYSCAT pour référencer à la place la vue SYSSTAT pouvant être mise à jour.

Le tableau suivant répertorie les vues SYSCAT qui contiennent des informations fédérées. Ces vues sont en lecture seule.

Tableau 89. Vues de catalogue généralement utilisées avec un système fédéré

Vues de catalogue	Description
SYSCAT.CHECKS	Contient les informations de contrainte que vous avez définies.
SYSCAT.COLCHECKS	Contient des colonnes référencées par une contrainte de vérification.
SYSCAT.COLUMNS	Contient des informations de colonne sur les objets de source de données (tables et vues) pour lesquels vous avez créé des pseudonymes.
SYSCAT.COLOPTIONS	Contient des informations sur les valeurs d'option de colonne que vous avez définies pour un pseudonyme.
SYSCAT.CONSTDEP	Contient la dépendance d'une contrainte informationnelle que vous avez définie.
SYSCAT.DATATYPES	Contient des informations de type de données sur les types de données DB2 locaux intégrés et définis par l'utilisateur.
SYSCAT.DBAUTH	Contient les droits de base de données détenus par des utilisateurs individuels et des groupes.
SYSCAT.FUNCMAPOPTIONS	Contient des informations sur les valeurs d'option que vous avez définies pour un mappage de fonctions.
SYSCAT.FUNCMAPPINGS	Contient les mappages de fonctions entre la base de données fédérée et les objets de source de données.
SYSCAT.INDEXCOLUSE	Contient les colonnes qui participent à un index.
SYSCAT.INDEXES	Contient les spécifications d'index pour les objets de source de données.

Tableau 89. Vues de catalogue généralement utilisées avec un système fédéré (suite)

Vues de catalogue	Description
SYSCAT.INDEXOPTIONS	Contient des informations sur les options d'index.
SYSCAT.KEYCOLUSE	Contient les colonnes qui participent à une clé définie par une contrainte de clé unique, de clé primaire ou de clé externe.
SYSCAT.NICKNAMES	Contient des informations sur les pseudonymes que vous avez créés.
SYSCAT.REFERENCES	Contient des informations sur les contraintes référentielles que vous avez définies.
SYSCAT.ROUTINES	Contient des fonctions DB2 locales définies par l'utilisateur ou des modèles de fonction. Les modèles de fonction sont utilisés pour se mapper à une fonction de source de données.
SYSCAT.REVTYPEMAPPINGS	Cette vue n'est pas utilisée. Tous les mappages de types de données sont enregistrés dans la vue SYSCAT.TYPEMAPPINGS.
SYSCAT.ROUTINEOPTIONS	Contient des informations sur les valeurs d'option de routine fédérées.
SYSCAT.ROUTINEPARMOPTIONS	Contient des informations sur les valeurs d'option des paramètres de routine fédérés.
SYSCAT.ROUTINEPARMS	Contient un paramètre ou le résultat d'une routine définie dans SYSCAT.ROUTINES.
SYSCAT.ROUTINESFEDERATED	Contient des informations sur les routines fédérées que vous avez définies.
SYSCAT.SERVERS	Contient les définitions de serveur que vous créez pour les serveurs de source de données.
SYSCAT.TABCONST	Chaque ligne représente une contrainte de table et de pseudonyme de type CHECK, UNIQUE, PRIMARY KEY ou FOREIGN KEY.
SYSCAT.TABLES	Contient des informations sur chaque table, vue fédérée et pseudonyme DB2 local que vous créez.
SYSCAT.TYPEMAPPINGS	Contient des mappages de types de données aval et des mappages de types de données inversés. Le mappage s'effectue vers les types de données DB2 locaux à partir des types de données de la source de données. Ces mappages sont utilisés lorsque vous créez un pseudonyme sur un objet de source de données.
SYSCAT.USEROPTIONS	Contient des informations sur les autorisations des utilisateurs que vous définissez lors de la création de mappages utilisateur entre la base de données fédérée et les serveurs de source de données.
SYSCAT.VIEWS	Contient des informations sur les vues fédérées locales que vous créez.

Tableau 89. Vues de catalogue généralement utilisées avec un système fédéré (suite)

Vues de catalogue	Description
SYSCAT.WRAPOPTIONS	Contient des informations sur les valeurs d'option que vous avez définies pour un encapsuleur.
SYSCAT.WRAPPERS	Contient le nom de l'encapsuleur et du fichier de bibliothèque pour chaque source de données pour laquelle vous créez un encapsuleur.

Le tableau suivant répertorie les vues SYSSTAT qui contiennent des informations fédérées. Ces vues, accessibles en lecture seule, contiennent des statistiques que vous pouvez mettre à jour.

Tableau 90. Vues de catalogue global fédérées pouvant être mises à jour

Vues de catalogue	Description
SYSSTAT.COLUMNS	Contient des informations statistiques sur chaque colonne des objets de source de données (tables et vues) pour lesquels vous avez créé des pseudonymes. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.
SYSSTAT.INDEXES	Contient des informations statistiques sur chaque spécification d'index pour des objets de source de données.
SYSSTAT.ROUTINES	Contient des informations statistiques sur chaque fonction définie par l'utilisateur. N'inclut pas de fonctions intégrées. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.
SYSSTAT.TABLES	Contient des informations sur chaque table de base. Les informations sur les vues, les synonymes et les alias ne sont pas incluses dans cette vue. Pour les tables basées sur un type structuré, seule la table racine d'une hiérarchie de tables est incluse dans la vue. Les statistiques ne sont pas enregistrées pour les colonnes héritées de tables basées sur un type structuré.

Chapitre 38. Options de mappage de fonctions pour les systèmes fédérés

Le serveur fédéré fournit des mappages par défaut entre les fonctions de DB2 et les fonctions de source de données. Pour la plupart des sources de données, les mappages de fonctions par défaut figurent dans les encapsuleurs. Pour utiliser une fonction de source de données non reconnue par le serveur fédéré ou pour modifier le mappage par défaut, vous créez un mappage de fonctions.

Lorsque vous créez un mappage de fonctions, vous spécifiez le nom de la fonction de source de données et devez activer la fonction mappée. Ensuite, lorsque vous utilisez la fonction mappée, l'optimiseur de requêtes compare le coût d'exécution de la fonction au niveau de la source de données au coût d'exécution de la fonction au niveau du serveur fédéré.

Tableau 91. Options pour les mappages de fonctions

Nom	Description
DISABLE	Permet d'activer ou de désactiver un mappage de fonctions par défaut. Les valeurs valides sont Y et N. La valeur par défaut est N.
REMOTE_NAME	Nom de la fonction de source de données. La valeur par défaut est le nom local.

Chapitre 39. Types de serveur valides dans les instructions SQL

Les types de serveur indiquent le type de source de données représenté par la définition de serveur.

Les types de serveur varient en fonction du fournisseur, de la finalité et du système d'exploitation. Les valeurs prises en charge dépendent de la source de données.

Pour la plupart des sources de données, vous devez spécifier une type de serveur valide dans l'instruction CREATE SERVER.

Tableau 92. Sources de données et types de serveur

Source de données	Type de serveur
BioRS	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Excel	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
IBM DB2 Universal Database for Linux, UNIX et Windows	DB2/UDB
IBM DB2 Universal Database for System i et AS/400	DB2/ISERIES
IBM DB2 Universal Database for z/OS	DB2/ZOS
IBM DB2 for VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (Obligatoire pour les sources de données JDBC prises en charge par les pilotes JDBC version 3.0 et ultérieure.)
Microsoft SQL Server	MSSQLSERVER (Obligatoire pour les sources de données prises en charge par le pilote DataDirect Connect ODBC 4.2 (ou version ultérieure) ou le pilote Microsoft SQL Server ODBC 3.0 (ou version ultérieure).)
ODBC	ODBC (Obligatoire pour les sources de données ODBC qui sont prises en charge par le pilote ODBC 3.x.)
OLE DB	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Oracle	ORACLE (Obligatoire pour les sources de données Oracle prises en charge par le logiciel client NET8.)
Sybase (CTLIB)	SYBASE
Fichiers structurés sous forme de table	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
Teradata	TERADATA

Tableau 92. Sources de données et types de serveur (suite)

Source de données	Type de serveur
services Web	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.
XML	Un type de serveur n'est pas requis dans l'instruction CREATE SERVER.

Chapitre 40. Mappages de type de données

Les mappages de types de données pour les sources de données non relationnelles incluent les mappages de types aval, les mappages de types inversés et les mappages de types propres à Unicode. Chaque source de données non relationnelle prend en charge des types de données spécifiques.

Mappages de types de données aval par défaut

Les mappages de type aval et inversé sont les deux types de mappage entre les types de données de sources de données et les types de données de bases de données fédérées. Un mappage de type aval est un mappage depuis un type de données distant vers un type de données local équivalent.

Vous pouvez remplacer un mappage de type de données par défaut, ou en créer un avec l'instruction CREATE TYPE MAPPING.

Ces mappages sont valides avec toutes les versions prises en charge, sauf mention contraire.

Pour tous les mappages de types de données aval par défaut d'une source de données vers une base de données fédérée, le schéma fédéré est SYSIBM.

Les tableaux suivants indiquent des mappages aval par défaut entre des types de données de bases de données fédérées et des types de données de sources de données.

Mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows.

Tableau 93. Mappages de types de données aval par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	-	-	-	-	-	DECFLOAT	-	0	-

Tableau 93. Mappages de types de données aval par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	Y	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	p	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. L'option de serveur SAME_DECFLT_ROUNDING est définie par défaut sur N et les opérations ne seront pas transférées sur la source de données distante à moins que l'option ne soit définie sur Y. Pour plus d'informations sur l'option de serveur SAME_DECFLT_ROUNDING, voir Référence des options de base de données DB2.

Mappages de types de données aval par défaut pour les sources de données DB2 for System i

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 for System i.

Tableau 94. Mappages de types de données aval par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N

Tableau 94. Mappages de types de données aval par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 pour VM et VSE.

Tableau 95. Mappages de types de données aval par défaut pour DB2 Server pour VM et VSE (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données DB2 for z/OS

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données DB2 for z/OS.

Tableau 96. Mappages de types de données aval par défaut de DB2 for z/OS (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
ROWID	-	-	-	-	Y	-	VARCHAR	40	-	Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

Mappages de types de données aval par défaut pour les sources de données Informix

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Informix.

Tableau 97. Mappages de types de données aval par défaut pour Informix(les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-

Tableau 97. Mappages de types de données aval par défaut pour Informix(les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	4	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DATETIME ²	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP(6)	10	6	-
DATETIME	6	10	11	15	-	-	TIMESTAMP(6)	10	6	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	130	-	-	-	-	DOUBLE	8	-	-
DECIMAL	1	32	255	255	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
INTERVAL	-	-	-	-	-	-	VARCHAR	25	-	-
INT8	-	-	-	-	-	-	BIGINT	19	0	-
LVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
MONEY	1	31	0	31	-	-	DECIMAL	-	-	-
MONEY	32	32	-	-	-	-	DOUBLE	8	-	-
NCHAR	1	254	-	-	-	-	CHARACTER	-	-	-
NCHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
NVARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SERIAL	-	-	-	-	-	-	INTEGER	4	-	-
SERIAL8	-	-	-	-	-	-	BIGINT	-	-	-
SMALLFLOAT	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TEXT	-	-	-	-	-	-	CLOB	2147483647	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-

Remarques :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Pour le type de données Informix DATETIME, le serveur fédéré de DB2 UNIX et Windows utilise le qualificatif de haut niveau Informix pour REMOTE_LENGTH et le qualificatif de bas niveau Informix pour REMOTE_SCALE.

Les qualificatifs Informix sont les constantes "TU_" définies dans le fichier datatime.h d'Informix Client SDK. Les constantes sont les suivantes :

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Mappages de types de données aval par défaut pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données JDBC.

Tableau 98. Mappages de types de données aval par défaut pour JDBC

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	Y
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
BINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
BIT	-	-	-	-	-	-	SMALLINT	2	-	-
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN-	-	-	-	-	-	-	SMALLINT	2	-	-
CHAR	-	254	-	-	-	-	CHAR	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
FLOAT	-	-	-	-	-	-	FLOAT	4	-	-
INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
LONGVARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
LONGVARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-
LONGVARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
LONGVARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
LONGNVARCHAR	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
LONGNVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCHAR ²	-	127	-	-	-	-	GRAPHIC	-	-	-
NCHAR ²	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
NCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
NCLOB ²	-	-	-	-	-	-	DBCLOB	-	-	-
NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
NVARCHAR ²	-	16336	-	-	-	-	VARGRAPHIC	-	-	-
NVARCHAR ²	16337	1073741823	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	4	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
TIME	-	-	-	-	-	-	TIME	3	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-

Tableau 98. Mappages de types de données aval par défaut pour JDBC (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
VARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	Y
VARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Les types de données suivants sont uniquement pris en charge par le pilote de périphérique JDBC 4.0 : NCHAR, NVARCHAR, LONGVARCHAR et NCLOB.

Les types de données suivants ne sont pas pris en charge par l'encapsuleur JDBC : DATALINK, OTHER, JAVA_OBJECT, DISTINCT, STRUCT, ARRAY et REF.

Mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server.

Tableau 99. Mappages de types de données aval par défaut pour Microsoft SQL Server

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
bigint ¹	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	Y
binary	255	8000	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	8	-	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	-	REAL	4	-	-
image	-	-	-	-	-	-	BLOB	2147483647	-	Y
int	-	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	127	-	-	-	-	CHAR	-	-	N
nchar	128	4000	-	-	-	-	VARCHAR	-	-	N
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	8	-	-
ntext	-	-	-	-	-	-	CLOB	2147483647	-	Y
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-

Tableau 99. Mappages de types de données aval par défaut pour Microsoft SQL Server (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires de données distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	8000	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_GUID	-	-	-	-	-	-	VARCHAR	-	-	Y
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	6	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	254	-	-	-	-	CHARACTER	-	-	N
SQL_WCHAR	255	8800	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

Remarque :

1. Ce mappage est uniquement valide avec Microsoft SQL Server 2000.

Mappages de types de données aval par défaut pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données ODBC.

Tableau 100. Mappages de types de données aval par défaut pour ODBC (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	8	-	-	-	-	FLOAT	8	-	-
SQL_FLOAT	-	4	-	-	-	-	FLOAT	4	-	-
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	2147483647	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	4	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TIMESTAMP(p)	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N
SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.

Mappages de types de données aval par défaut pour les sources de données Oracle NET8

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Oracle NET8.

Tableau 101. mappages de types de données aval par défaut Oracle NET8

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	\0	CHAR	0	0	N
CHAR	255	2000	0	0	-	\0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	\0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	\0	TIMESTAMP(6)	0	0	N
FLOAT	1	126	0	0	-	\0	DOUBLE	0	0	N
LONG	0	0	0	0	-	\0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	\0	BLOB	2147483647	0	Y
NUMBER	10	18	0	0	-	\0	BIGINT	0	0	N
NUMBER	1	38	-84	127	-	\0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	4	0	0	-	\0	SMALLINT	0	0	N
NUMBER	5	9	0	0	-	\0	INTEGER	0	0	N
NUMBER	-	10	0	0	-	\0	DECIMAL	0	0	N
RAW	1	2000	0	0	-	\0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	\0	CHAR	18	0	N
TIMESTAMP(p) ¹	-	-	-	-	-	\0	TIMESTAMP(6)	10	6	N
VARCHAR2	1	4000	0	0	-	\0	VARCHAR	0	0	N

Remarque :

1.

- TIMESTAMP(p) représente un horodatage avec une échelle de variable comprise entre 0 et 9. L'échelle de l'horodatage Oracle est mappé vers TIMESTAMP(6) par défaut. Vous pouvez modifier ce mappage de type par défaut et mapper l'horodatage Oracle TIMESTAMP vers un horodatage TIMESTAMP fédéré de la même échelle en utilisant un mappage de type défini par l'utilisateur.
- Ce mappage est uniquement valide pour les configurations client et de serveurs Oracle 9i (ou ultérieur).

Mappages de types de données aval par défaut pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Sybase.

Tableau 102. Mappages de types de données aval par défaut pour Sybase CTLIB

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
binary	1	254	-	-	-	-	CHAR	-	-	Y
binary	255	32672	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-

Tableau 102. Mappages de types de données aval par défaut pour Sybase CTLIB (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (voir varchar)										
date	-	-	-	-	-	-	DATE	-	-	-
date	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
datetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
datetimn	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	32672	-	-	-	-	VARCHAR	-	-	N
nchar null (voir nvarchar)										
numeric	1	31	0	31	-	-	DECIMAL	-	-	-
numeric	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
nvarchar	1	32672	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	-	-	-	-	-	-	VARCHAR	30	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
time	-	-	-	-	-	-	TIME	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N

Tableau 102. Mappages de types de données aval par défaut pour Sybase CTLIB (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
unichar null (voir univarchar)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	Y
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.
2. Valide pour les bases de données fédérées non Unicode.

Mappages de types de données aval par défaut pour les sources de données Teradata

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Teradata.

Tableau 103. Mappages de types de données aval par défaut pour Teradata (les colonnes ne sont pas toutes représentées)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	Y
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	Y
BYTE	32673	64000	-	-	-	-	BLOB	-	-	-
BYTEINT	-	-	-	-	-	-	SMALLINT	-	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	64000	-	-	-	-	CLOB	-	-	-
CLOB	1	2097088000 (Latin)	-	-	-	-	CLOB	-	-	-
CLOB	1	1048544000 (Unicode)	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DOUBLE PRECISION	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	-	-
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	-	-
GRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
INTERVAL	-	-	-	-	-	-	CHAR	-	-	-
NUMERIC	1	18	0	18	-	-	DECIMAL	-	-	-

Tableau 103. Mappages de types de données aval par défaut pour Teradata (les colonnes ne sont pas toutes représentées) (suite)

Type de nom distant	Réseau Len inférieur distant	Réseau Len supérieur distant	Echelle inférieure distante	Echelle supérieure distante	Données binaires distantes	Opérateurs de données distantes	Nom de type fédéré	Longueur fédérée	Echelle fédérée	Données binaires fédérées
REAL	-	-	-	-	-	-	DOUBLE	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	0	21	0	21	-	-	TIME	-	-	-
TIMESTAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP(6)	10	6	-
VARBYTE	1	32762	-	-	-	-	VARCHAR	-	-	Y
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

Remarque :

1. Le type fédéré est TIMESTAMP(0) si le paramètre de configuration date_compat est défini sur ON.

Echantillons de mappages de types de données aval

Vous pouvez utiliser les exemples de mappages de types aval pour exploiter la prise en charge du type de données TIMESTAMP.

Pour les sources de données Informix, ces mappages sont utilisés pour les colonnes de pseudonymes, les paramètres des procédures fédérées, les passes-systèmes et les ensembles de résultats des procédures fédérées.

Pour les autres sources de données, ces mappages affectent uniquement les colonnes de pseudonymes et les paramètres des procédures fédérées. Les passe-systèmes et les ensembles de résultats des procédures fédérées ne sont pas concernés.

Mappages de types de données aval - exemple pour Informix

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Informix.

Vous devez créer ces mappages avant les objets fédérés.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,10);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,11);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,12);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,13);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,14);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE informix REMOTE TYPE datetime(0,15);
```

Mappages de types de données aval - exemple pour Microsoft SQL Server

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Microsoft SQL Server.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE mssqlserver REMOTE TYPE "datetime";

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE mssqlserver REMOTE TYPE "smalldatetime";
```

Mappages de types de données aval - exemple pour Oracle

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Oracle.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(5);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(7)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(7);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(8)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(8);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(9)
  TO SERVER TYPE oracle REMOTE TYPE timestamp(9);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE oracle REMOTE TYPE date;
```

Mappages de types de données aval - exemple pour Sybase

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Sybase.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE sybase REMOTE TYPE datetime);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE sybase REMOTE TYPE smalldatetime);
```

Mappages de types de données aval - exemple pour Teradata

Lorsque vous créez des objets fédérés, vous pouvez utiliser l'exemple de mappage de types aval fourni pour Teradata.

Vous devez créer ces mappages avant un pseudonyme ou une procédure fédérée.

```
CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(0)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(0);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(1)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(1);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(2)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(2);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(3)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(3);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(4)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(4);

CREATE TYPE MAPPING FROM LOCAL TYPE timestamp(5)
  TO SERVER TYPE teradata REMOTE TYPE timestamp(5);
```

Mappages de types de données inversés par défaut

Pour la plupart des sources de données, les mappages de type par défaut figurent dans les encapsuleurs.

Les mappages de type aval et inversé sont les deux types de mappage entre les types de données de sources de données et les types de données de bases de données fédérées. Un mappage de type aval est un mappage depuis un type de données distant vers un type de données local équivalent. Un mappage de type inversé est un mappage utilisé avec le langage DDL transparent pour créer ou modifier des tables distantes.

Les mappages de type par défaut pour les sources de données DB2 se trouvent dans l'encapsuleur DRDA. Les mappages de type par défaut pour Informix se trouvent dans l'encapsuleur INFORMIX, et ainsi de suite.

Lorsque vous définissez une table ou vue distante d'une base de données fédérée, la définition inclut un mappage de type inversé. Le mappage s'effectue depuis un type de données de base de données fédérée locale pour chacune des colonnes, et le type de données distantes correspondantes. Par exemple, il y a un mappage de type inversé par défaut où le type de données local REAL pointe sur le type SMALLFLOAT d'Informix.

Les bases de données fédérées ne prennent pas en charge le mappage pour les types LONG VARCHAR, LONG VARGRAPHIC et les types définis par l'utilisateur.

Lorsque vous créez une table distante avec l'instruction CREATE TABLE, vous devez indiquer les types de données locales que vous voulez inclure dans la table distante. Le mappage de type inversé par défaut va attribuer à ces colonnes les types distants correspondants. Supposons que vous utilisiez l'instruction CREATE TABLE pour définir une table Informix avec une colonne C2. Vous indiquez BIGINT comme type de données pour C2 dans l'instruction. Le mappage de type inversé par défaut du type BIGINT dépend de la version d'Informix sur laquelle vous créez votre table. Le mappage de C2 dans la table Informix sera effectué vers le type DECIMAL si vous utilisez Informix version 8 et vers le type INT8 si vous utilisez Informix version 9.

Vous pouvez remplacer un mappage de type de données inversé par défaut, ou en créer un avec l'instruction CREATE TYPE MAPPING.

Les tableaux suivants indiquent des mappages inversés par défaut entre des types de données locales de bases de données fédérées et des types de données de sources de données distantes.

Ces mappages sont valides avec toutes les versions prises en charge, sauf mention contraire.

Mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données d'une base de données DB2 pour Linux, UNIX et Windows.

Tableau 104. Mappages de types de données inversés par défaut pour une base de données DB2 pour Linux, UNIX, et Windows (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires fédérées
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	8	-	-	-	-	DECFLOAT	-	0	-
DECFLOAT ²	-	16	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(<i>p</i>)	-	-	<i>p</i>	<i>p</i>	-	-	TIMESTAMP(<i>p</i>)	-	<i>p</i> ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Remarque :

- Lorsque le paramètre date_compat est défini sur OFF, le type fédéré DATE est mappé vers TIMESTAMP(0).
- L'option de serveur SAME_DECFLT_ROUNDING est définie par défaut sur N et les opérations ne seront pas transférées sur la source de données distante à moins que l'option ne soit définie sur Y. Pour plus d'informations sur l'option de serveur SAME_DECFLT_ROUNDING, voir Référence des options de base de données DB2.
- Pour la version 9.5, ou une version antérieure, l'échelle distante pour TIMESTAMP est 6.

Mappages de types de données inversés par défaut pour les sources de données DB2 for System i

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 for System i.

Tableau 105. Mappages de types de données inversés par défaut pour DB2 for System i (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données binaires	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	Y	-	CHARACTER	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

Mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 pour VM et VSE.

Tableau 106. Mappages de types de données inversés par défaut pour DB2 pour VM et VSE (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N

Tableau 106. Mappages de types de données inversés par défaut pour DB2 pour VM et VSE (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

Mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données DB2 for z/OS.

Tableau 107. Mappages de types de données inversés par défaut pour DB2 for z/OS (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Mappages de types de données inversés par défaut pour les sources de données Informix

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Informix.

Tableau 108. Mappages de types de données inversés par défaut pour Informix

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2147483647	-	-	-	-	TEXT	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	DATETIME	6	10	-
TIMESTAMP	-	10	-	-	-	-	DATETIME	0	15	-
VARCHAR	1	254	-	-	N	-	VARCHAR	-	-	-
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	Y	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

Remarque :

1. Ce mappage est uniquement valide avec la version 8 (ou une version antérieure) d'Informix.
2. Ce mappage est uniquement valide avec la version 9 (ou une version ultérieure) d'Informix.

Pour le type de données Informix DATETIME, le serveur fédéré utilise le qualificatif de haut niveau Informix pour REMOTE_LENGTH et le qualificatif de bas niveau Informix pour REMOTE_SCALE.

Les qualificatifs Informix sont les constantes "TU_" définies dans le fichier datatype.h d'Informix Client SDK. Les constantes sont les suivantes :

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)
6 = HOUR	12 = FRACTION(2)	

Mappages de types de données inversés par défaut pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données JDBC qui répondent aux mappages de types du pilote JDBC DB2.

Tableau 109. Mappages de types de données inversés par défaut pour JDBC

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-

Tableau 109. Mappages de types de données inversés par défaut pour JDBC (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
CHAR	-	-	-	-	Y	-	BINARY	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	NCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	NCHAR	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP(p)	-	min(9,p)	-
VARCHAR	-	-	-	-	Y	-	VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	-	-	NVARCHAR	-	-	-

Mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server.

Tableau 110. Mappages de types de données inversés par défaut pour Microsoft SQL Server (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-

Tableau 110. Mappages de types de données inversés par défaut pour Microsoft SQL Server (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
VARCHAR	1	8000	-	-	Y	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

Remarque :

1. Ce mappage est uniquement valide avec Microsoft SQL Server 2000.

Mappages de types de données inversés par défaut pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données ODBC.

Tableau 111. Mappages de types de données inversés par défaut pour ODBC (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	SQL_BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_LONGVARBINARY	-	-	-
CHAR	-	-	-	-	Y	-	SQL_BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_CHAR	-	-	-
CLOB	-	-	-	-	-	-	SQL_LONGVARCHAR	-	-	-
DATE	-	4	-	-	-	-	SQL_TYPE_DATE	-	-	-
DBCLOB	-	-	-	-	-	-	SQL_WLONGVARCHAR	-	-	-
DECIMAL	-	-	-	-	-	-	SQL_DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	SQL_DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	SQL_FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	SQL_WCHAR	-	-	-
INTEGER	-	-	-	-	-	-	SQL_INTEGER	-	-	-
REAL	-	4	-	-	-	-	SQL_REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SQL_SMALLINT	-	-	-
TIME	-	3	-	-	-	-	SQL_TYPE_TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	SQL_TYPE_TIMESTAMP(p)	-	-	-
VARCHAR	-	-	-	-	Y	-	SQL_VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	Y	-	SQL_WVARCHAR	-	-	-

Mappages de types de données inversés par défaut pour les sources de données Oracle NET8

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Oracle NET8.

Tableau 112. Mappages de types de données inversés par défaut pour Oracle NET8

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	Y	\0	BLOB	0	0	Y
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	Y	\0	RAW	0	0	Y
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N
DATE ¹	0	4	0	0	N	\0	DATE	0	0	N
DECIMAL	0	0	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	8	0	0	N	\0	NUMBER	0	0	N
DECFLOAT	0	16	0	0	N	\0	NUMBER	0	0	N
DOUBLE	0	8	0	0	N	\0	FLOAT	126	0	N
FLOAT	0	8	0	0	N	\0	FLOAT	126	0	N
INTEGER	0	4	0	0	N	\0	NUMBER	10	0	N
REAL	0	4	0	0	N	\0	FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0	NUMBER	5	0	N
TIME	0	3	0	0	N	\0	DATE	0	0	N
TIMESTAMP ²	0	10	0	0	N	\0	DATE	0	0	N
TIMESTAMP(p) ³	-	-	-	-	N	\0	TIMESTAMP(p)-	-	-	N
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	Y	\0	RAW	0	0	Y

Remarque :

1. Lorsque le paramètre date_compat est défini sur OFF, le type fédéré DATE est mappé vers la date Oracle. Lorsque le paramètre date_compat est défini sur ON, le type fédéré DATE (équivalent à TIMESTAMP(0)) est mappé vers Oracle TIMESTAMP(0).
2. Ce mappage est uniquement valide avec Oracle Version 8.
3.
 - TIMESTAMP(p) représente un horodatage avec une échelle de variable comprise entre 0 et 9 pour Oracle et 0 et 12 pour la fédération. Lorsque l'échelle va de 0 à 9, le type TIMESTAMP distant d'Oracle a la même échelle que le type TIMESTAMP fédéré. Si l'échelle du type TIMESTAMP fédéré est supérieure à 9, l'échelle correspondante du type TIMESTAMP d'Oracle est 9, qui correspond à l'échelle d'Oracle la plus élevée.
 - Ce mappage est uniquement valide avec Oracle Version 9, 10 et 11.

Mappages de types de données inversés par défaut pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Sybase.

Tableau 113. Mappages de types de données inversés par défaut pour Sybase CTLIB

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-

Tableau 113. Mappages de types de données inversés par défaut pour Sybase CTLIB (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR ¹	1	255	-	-	N	-	varchar	-	-	-
VARCHAR ¹	256	32672	-	-	N	-	text	-	-	-
VARCHAR ²	1	16384	-	-	N	-	varchar	-	-	-
VARCHAR ²	16385	32672	-	-	N	-	text	-	-	-
VARCHAR ¹	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	Y	-	image	-	-	-
VARCHAR ²	1	16384	-	-	Y	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	Y	-	image	-	-	-

Remarque :

1. Ce mappage est uniquement valide pour CTLIB avec la version 12.0 (ou une version antérieure) de Sybase.
2. Ce mappage est uniquement valide pour CTLIB avec la version 12.5 (ou une version ultérieure) de Sybase.

Mappages de types de données inversés par défaut pour les sources de données Teradata

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Teradata.

Tableau 114. Mappages de types de données inversés par défaut pour Teradata (les colonnes ne sont pas toutes représentées)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2097088000	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DBCLOB ¹	1	64000	-	-	-	-	VARGRAPHIC	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DECIMAL	19	31	0	31	-	-	FLOAT	8	-	-
DOUBLE	-	-	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	FLOAT	8	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-

Tableau 114. Mappages de types de données inversés par défaut pour Teradata (les colonnes ne sont pas toutes représentées) (suite)

Nom de type fédéré	Réseau Len inférieur fédéré	Réseau Len supérieur fédéré	Echelle inférieure fédérée	Echelle supérieure fédérée	Données binaires fédérées	Opérateurs de données fédérées	Type de nom distant	Longueur distante	Echelle distante	Données binaires distantes
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Remarque :

1. Le type de données Teradata VARGRAPHIC peut comprendre uniquement la longueur indiquée (1 à 32 000) d'un type d'objet DBCLOB.

Mappages de types de données par défaut Unicode

Certaines sources de données prennent en charge les mappages de types de données aval et les mappages de types de données inversés pour les bases de données Unicode.

Mappages de types de données aval par défaut Unicode pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données JDBC avec une base de données fédérée Unicode.

Tableau 115. Mappages de types de données aval par défaut Unicode pour les sources de données JDBC

UTF-8	JDBC	
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	VARCHAR	1 à 32 672 octets
CLOB	CLOB	-
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NVARCHAR	1 à 16 336 caractères
DBCLOB	NCLOB	-

Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données JDBC avec une base de données fédérée Unicode.

Tableau 116. Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC

UTF-8	JDBC	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR

Tableau 116. Mappages de types de données inversés par défaut Unicode pour les sources de données JDBC (suite)

UTF-8		JDBC
Type de données	Longueur	Type de données
VARCHAR	1 à 32 672 octets	VARCHAR
CLOB	1 à 2 147 483 647 octets	CLOB
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 16 336 caractères	NVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	NCLOB

Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Microsoft SQL Server avec une base de données fédérée Unicode.

Tableau 117. Mappages de types de données aval par défaut Unicode pour les sources de données Microsoft SQL Server

UTF-8		Microsoft SQL Server
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	CHAR	255 à 8 000 octets
	VARCHAR	1 à 8 000 octets
CLOB	TEXT	-
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NCHAR	128 à 16 336 caractères
	NVARCHAR	1 à 16 336 caractères
DBCLOB	NTEXT	-

Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Microsoft SQL Server avec une base de données fédérée Unicode.

Tableau 118. Mappages de types de données inversés par défaut Unicode pour les sources de données Microsoft SQL Server

UTF-8		Microsoft SQL Server
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR
VARCHAR	1 à 32 672 octets	VARCHAR
CLOB	1 à 2 147 483 647 octets	TEXT
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 16 336 caractères	NVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	NTEXT

Mappages de types de données aval par défaut Unicode pour les sources de données NET8

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données NET8 avec une base de données fédérée Unicode.

Tableau 119. Mappages de types de données aval par défaut Unicode pour les sources de données NET8

UTF-8		Oracle
Type de données	Type de données	Longueur
CHAR	CHAR	1 à 254 octets
VARCHAR	CHAR	255 à 2 000 octets
	VARCHAR2	1 à 4 000 octets
DBCLOB	NCLOB	
GRAPHIC	NCHAR	1 à 127 caractères
VARGRAPHIC	NCHAR	128 à 1 000 caractères
	NVARCHAR2	1 à 2 000 caractères

Mappages de types de données inversés par défaut Unicode pour les sources de données NET8

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données NET8 avec une base de données fédérée Unicode.

Tableau 120. Mappages de types de données inversés par défaut Unicode pour les sources de données NET8

UTF-8		Oracle
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	CHAR
VARCHAR	1 à 4 000 octets	VARCHAR2
CLOB	1 à 2 147 483 647 octets	CLOB
GRAPHIC	1 à 127 caractères	NCHAR
VARGRAPHIC	1 à 2 000 caractères	NVARCHAR2
DBCLOB	1 à 1 073 741 823 caractères	NCLOB

Mappages de types de données aval par défaut Unicode pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données ODBC avec une base de données fédérée Unicode.

Tableau 121. Mappages de types de données aval par défaut Unicode pour les sources de données ODBC

UTF-8		ODBC
Type de données	Type de données	Longueur
CHAR	SQL_CHAR	1 à 254 octets

Tableau 121. Mappages de types de données aval par défaut Unicode pour les sources de données ODBC (suite)

UTF-8	ODBC	
Type de données	Type de données	Longueur
VARCHAR	SQL_CHAR	255 à 32 672 octets
	SQL_VARCHAR	1 à 32 672 octets
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	1 à 127 caractères
VARGRAPHIC	SQL_WCHAR	128 à 16 336 caractères
	SQL_WVARCHAR	1 à 16 336 caractères
DBCLOB	SQL_WLONGVARCHAR	-

Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données ODBC avec une base de données fédérée Unicode.

Tableau 122. Mappages de types de données inversés par défaut Unicode pour les sources de données ODBC

UTF-8	ODBC	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	SQL_CHAR
VARCHAR	1 à 32 672 octets	SQL_VARCHAR
CLOB	1 à 2 147 483 647 octets	SQL_LONGVARCHAR
GRAPHIC	1 à 127 caractères	SQL_WCHAR
VARGRAPHIC	1 à 16 336 caractères	SQL_WVARCHAR
DBCLOB	1 à 1 073 741 823 caractères	SQL_WLONGVARCHAR

Mappages de types de données aval par défaut Unicode pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données aval par défaut pour les sources de données Sybase CTLIB avec une base de données fédérée Unicode.

Tableau 123. Mappages de types de données aval par défaut Unicode pour les sources de données Sybase CTLIB

UTF-8	Sybase	
Type de données	Type de données	Longueur
CHAR	char	1 à 254 octets
	nchar	1 à 127 caractères
VARCHAR	char	255 à 32 672 octets
	varchar	1 à 32 672 octets
	nchar	128 à 16 336 caractères
	nvarchar	1 à 16 336 caractères

Tableau 123. Mappages de types de données aval par défaut Unicode pour les sources de données Sybase CTLIB (suite)

UTF-8	Sybase	
Type de données	Type de données	Longueur
CLOB	text	
GRAPHIC	unichar	1 à 127 caractères
VARGRAPHIC	unichar	128 à 16 336 caractères
	univarchar	1 à 16 336 caractères

Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase

Le tableau suivant répertorie les mappages de types de données inversés par défaut pour les sources de données Sybase CTLIB avec une base de données fédérée Unicode.

Tableau 124. Mappages de types de données inversés par défaut Unicode pour les sources de données Sybase CTLIB

UTF-8	Sybase	
Type de données	Longueur	Type de données
CHAR	1 à 254 octets	char
VARCHAR	1 à 32 672 octets	varchar
CLOB	1 à 2 147 483 647 octets	text
GRAPHIC	1 à 127 caractères	unichar
VARGRAPHIC	1 à 16 336 caractères	univarchar

Types de données pris en charge pour les sources de données non relationnelles

Pour la plupart des sources de données non relationnelles, vous devez remplir les informations contenues dans les colonnes, y compris le type de données, lorsque vous créez les pseudonymes pour accéder à la source de données.

Certains encapsuleurs non relationnels créent toutes les colonnes nécessaires pour accéder à une source de données. On appelle ces colonnes les *colonnes fixes*. Avec d'autres encapsuleurs, vous pourrez indiquer certains types de données, voire tous, pour les colonnes dans l'instruction CREATE NICKNAME.

Les sections suivantes, répertorient les encapsuleurs pour lesquels vous pouvez indiquer les types de données, et les types de données pris en charge par les encapsuleurs.

Types de données pris en charge par l'encapsuleur BioRS

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur BioRS.

Tableau 125. Types de données BioRS mappés vers les types de données DB2

Types de données BioRS	Type de données DB2
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

La longueur maximum autorisée pour le type de données CLOB est de 5 mégaoctets.

Types de données pris en charge par l'encapsuleur Excel

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Excel.

Tableau 126. Types de données Excel mappés vers les types de données DB2

Types de données Excel	Type de données DB2
character	CHARACTER
date	DATE
number	DOUBLE
number	FLOAT
integer	INTEGER
character	VARCHAR

Types de données pris en charge par l'encapsuleur Script

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Script.

Tableau 127. Types de données Script mappés vers les types de données DB2

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA

Types de données pris en charge par l'encapsuleur Fichiers structurés sous formes de tables

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur Fichiers structurés sous formes de tables.

Tableau 128. Types de données Fichiers structurés sous formes de tables mappés vers les types de données DB2

Types de données Fichiers structurés sous formes de tables	Type de données DB2
character	CHARACTER
character	CLOB (longueur maximum : 5 mégaoctets)
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR

Types de données pris en charge par l'encapsuleur services Web

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur services Web. L'encapsuleur services Web utilise les types de données XML.

Tableau 129. Types de données XML mappés vers les types de données DB2 pour l'encapsuleur services Web

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA

Types de données pris en charge par l'encapsuleur XML

Le tableau suivant répertorie les types de données DB2 pris en charge par l'encapsuleur XML.

Tableau 130. Types de données XML mappés vers les types de données DB2 pour l'encapsuleur XML

Types de données XML	Type de données DB2
character	BLOB
character	CHARACTER
character	CHARACTER FOR BIT DATA
character	CLOB (longueur maximum : 5 mégaoctets)
date	DATE
number	DECIMAL
number	DOUBLE
number	FLOAT
integer	INTEGER
number	REAL
integer	SMALLINT
character	VARCHAR
character	VARCHAR FOR BIT DATA
Document XML	XML

Documentation du produit

La documentation est disponible sur différents supports et dans plusieurs formats, notamment dans l'aide qui s'affiche directement dans l'interface du produit, dans le centre de documentation de la suite et dans des documents au format PDF.

Vous pouvez également commander en ligne des publications IBM au format papier ou via votre représentant IBM local.

Pour commander des publications en ligne, visitez le Centre de publications IBM à l'adresse www.ibm.com/shop/publications/order.

Vous pouvez envoyer vos commentaires sur la documentation en suivant les procédures ci-après :

- Formulaire de commentaire en ligne : www.ibm.com/software/data/rcf/
- Courriel : comments@us.ibm.com

Lecture des diagrammes de syntaxe

Les règles suivantes s'appliquent aux diagrammes de syntaxe utilisés dans ce document :

- Lisez les diagrammes de syntaxe de gauche à droite, de haut en bas en suivant le chemin de la ligne. Les conventions suivantes sont utilisées :
 - Le symbole >>--- indique le début d'un diagramme de syntaxe.
 - Le symbole ---> indique que le diagramme de syntaxe continue sur la ligne suivante.
 - Le symbole >--- indique qu'un diagramme de syntaxe a été commencé sur la ligne précédente.
 - Le symbole --->< indique la fin du diagramme de syntaxe.
- Les éléments obligatoires apparaissent sur la ligne horizontale (chemin principal).



- Les éléments facultatifs apparaissent sous le chemin principal.



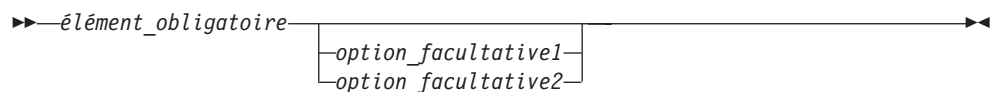
Si un élément facultatif apparaît au-dessus du chemin principal, il n'a pas d'incidence sur l'exécution de l'élément de syntaxe et est utilisé uniquement pour faciliter la lecture des données.



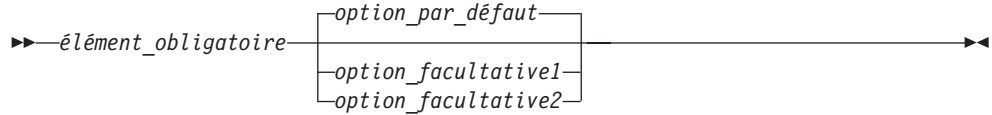
- Si vous pouvez faire une sélection parmi plusieurs éléments, ceux-ci apparaissent à la verticale, dans une pile.
Si vous devez sélectionner l'un des éléments, un seul élément de la pile apparaît dans le chemin principal.



Si la sélection de l'un des éléments est facultative, l'ensemble de la pile apparaît sous le chemin principal.



Si l'un des éléments est l'élément par défaut, il apparaît au-dessus du chemin principal et les options restantes sont affichées ci-dessous.



- Une flèche orientée à gauche, au-dessus de la ligne principale, indique un élément qui peut être répété.

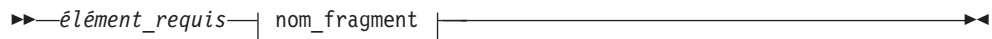


Si la flèche de répétition contient une virgule, vous devez séparer les éléments répétés avec une virgule.



Une flèche de répétition au-dessus d'une pile indique que vous pouvez répéter les éléments de la pile.

- Un diagramme peut parfois être divisé en fragments. Le fragment de syntaxe est affiché séparément du diagramme de syntaxe principal mais le contenu du fragment doit être lu comme s'il se trouvait dans le chemin principal du diagramme.



Nom-Fragment :



- Les mots clés et les abréviations minimales associées apparaissent en majuscules. Vous devez les orthographier correctement.
- Les variables apparaissent en majuscules et en italique (par exemple, *column-name*). Elles représentent les noms ou les valeurs définis par l'utilisateur.
- Séparez les mots clés et les paramètres par au moins un espace si aucun signe de ponctuation n'apparaît dans le diagramme.
- Entrez les signes de ponctuation, les parenthèses, les opérateurs arithmétiques et les autres symboles tels qu'ils sont indiqués dans le diagramme.
- Les notes de pied de page sont indiquées par un numéro entre parenthèses, par exemple (1).

Documentation accessible

La documentation est fournie au format XHTML. Vous pouvez la consulter dans la plupart des navigateurs Web.

XHTML vous permet de visualiser la documentation en fonction de préférences d'affichage définies dans votre navigateur. Cela vous permet d'utiliser des lecteurs d'écran ainsi que d'autres technologies d'assistance.

Les diagrammes de syntaxe sont livrés en notation décimale à points. Ce format est uniquement disponible si vous accédez à la documentation en ligne à l'aide d'un lecteur d'écran.

Remarques

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.

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 à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

Le paragraphe ci-dessous 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 CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

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. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. 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.

Ces informations sont fournies uniquement à titre de planification. Elles sont susceptibles d'être modifiées avant la mise à disposition des produits décrits.

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.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programme sont fournis "EN L'ETAT", sans garantie aucune. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation de ces exemples de programmes.

Toute copie totale ou partielle de ces exemples de programmes et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _entrez l'année ou les années_. All rights reserved.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

Les marques IBM et certains marques non IBM sont repérées lors de leur première apparition dans ce document avec le symbole adéquat.

IBM, le logo IBM et ibm.com sont des marques ou des marques déposées d'International Business Machines Corp., enregistrées dans de nombreuses juridictions internationales. Les noms de produits ou de services autres peuvent être des marques d'IBM ou de sociétés autres. Une liste à jour des marques d'IBM est disponible sur le Web dans "Copyright and trademark information" à l'adresse suivante : www.ibm.com/legal/copytrade.shtml.

Les termes qui suivent sont des marques d'autres sociétés :

Adobe, le logo Adobe, PostScript, le logo PostScript sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

IT Infrastructure Library est une marque de The Central Computer and Telecommunications Agency, qui fait désormais partie de The Office of Government Commerce.

Intel, le logo Intel, Intel Inside, le logo Intel Inside, Intel Centrino, le logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, et Pentium sont des marques d'Intel Corporation ou de ses filiales aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds 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.

ITIL est une marque déposée et une marque déposée communautaire de l'Office of Government Commerce et est déposée auprès du Patent and Trademark Office américain.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Cell Broadband Engine est une marque de Sony Computer Entertainment, Inc. aux Etats-Unis et/ou dans certains autres pays. Elle est utilisée sous licence.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

A

- accès aux données
 - vues fédérées 163
- accès aux données avec des pseudonymes 156
- accessibilité 475
- Accessibilité 471
- activation
 - validation en deux phases fédérées 125
- alias
 - accès aux données 156, 160
 - accès aux sources de données 205
 - commande EXPORT, restrictions 153
 - commande EXPORT, utilisation 151
 - commande IMPORT, exemples 152
 - commande IMPORT, restrictions 151
 - commande IMPORT, utilisation 151
 - contraintes 140
 - contraintes informationnelles 169, 170
 - contraintes informationnelles sur les pseudonymes 169
 - création 161
 - déclencheurs 156
 - description 15
 - données XML 187
 - instructions SQL 157
 - langage XQuery 187
 - mise à jour des statistiques 181
 - mise en mémoire cache 321
 - objets de source de données valides 16
 - objets éloignés et locaux 155
 - Oracle Label Security 174
 - pseudonyme sur un pseudonyme 165
 - statistiques 176, 180
 - collecte automatique 184
 - plusieurs pseudonymes 178
 - pseudonyme unique 179
 - statut, mises à jour
 - Centre de contrôle DB2 181
 - syntaxe WITH HOLD 156
 - tables de mémoire cache 291
- ALTER NICKNAME (instruction)
 - exemple
 - type de données local 56
- ALTER WRAPPER (instruction) 24
- analyse de transfert de l'exécution sur la base de données
 - caractéristiques de pseudonymes, incidence sur les 242
 - caractéristiques de requête, incidence sur les 244
 - caractéristiques de serveur, incidence sur les 235
 - description 11, 234
 - prédicats avec des modèles de fonction 249

- applications
 - pseudonymes dans 205
- architecture
 - validation en deux phases fédérées 116
- assistant Table de mémoire cache 293
 - assistants 293
- atomicité 122
 - conservation dans des instructions 141
- attributions
 - fédéré 144
- authentification de proxy Oracle 352

B

- Base de données DB2
 - fonctions fédérées prises en charge 353
 - options d'encapsuleur 361
 - options de colonne 361
 - options de mappage utilisateur 361
 - options de pseudonyme 361
 - options de serveur 361
- base de données fédérée
 - encapsuleurs 6
 - modules d'encapsuleur 6
 - objets locaux 155
- bases de données fédérées
 - catalogue système 10
 - description 6
- BioRS
 - fonctions fédérées prises en charge 353
 - options d'encapsuleur 357
 - options de colonne 357
 - options de mappage utilisateur 357
 - options de pseudonyme 357
 - options de serveur 357
 - types de données, pris en charge 466

C

- catalogue
 - Voir catalogue global 431
- catalogue, outils 181
- catalogue d'outils, DB2 181
- catalogue global 48
 - description 10
 - mise à jour des statistiques 249
 - vues contenant des informations fédérées 431
- catalogue local
 - Voir catalogue global 10
- Centre de commande
 - utilisation fédérée 7
- Centre de contrôle
 - interface pour systèmes fédérés 7
- Centre de contrôle DB2 8
- Centre de santé
 - indicateur de santé 197
 - instantané de santé 199
- chaînes
 - séquences de classement 19
 - vide, utilisé avec Oracle 146
- chiffrement
 - description 299
- clause de demande de verrouillage 214
- clause RETURN DATA UNTIL 194
- CLP (Command Line Processor)
 - fonctions fédérées 7
- COMM_RATE (option de serveur)
 - optimisation globale, incidence sur l' 275
- Commande DB2FEDGENTF
 - exemples 98
 - syntaxe 98
- commande EXPORT
 - pseudonyme, utilisation 153
- Commande EXPORT
 - pseudonyme, utilisation 151
- commande IMPORT
 - pseudonymes, exemples 152
- Commande IMPORT
 - pseudonyme, utilisation 151
 - pseudonymes, restrictions 151
- commutateur du moniteur d'horodatage 285
- commutateurs du contrôleur fédéré 285
- commutateurs du contrôleur système fédéré 285
- compatible VARCHAR2 236
- compensation, description 12
- configuration
 - validation en deux phases fédérées 125
- configuration requise 127
- configuration requise pour la source de données lors de la validation en deux phases fédérées 127
- connexions accréditées fédérées
 - API 304, 308
 - application échantillon 308
 - avantages 302
 - conception d'application 304
 - configuration 306, 310, 313
 - connexions accréditées fédérées
 - connexions entrantes 305
 - connexions sortantes 305
 - utilisateurs de proxy 313
 - connexions accréditées sortantes 313
 - connexions de bout en bout 303, 306, 310
 - connexions sortantes accréditées 303
- création
 - contextes accrédités 306, 310, 313
 - décrit 301, 303
 - définitions de serveur 313

connexions accréditées fédérées (*suite*)
 exigences d'ID utilisateur 306, 310, 313
 exigences de mot de passe 306, 310, 313
 explicite 301
 implicite
 connexions accréditées fédérées 301
 mappages utilisateur 313, 318
 requirements de mappage utilisateur 306, 310
 scénarios 305, 306, 308, 310, 313
 types 303
 utilisateurs de proxy fédéré décrit 313
 contextes accrédités décrit 301
 contextes accrédités fédérés
 authentification de proxy Oracle 352
 sources de données prises en charge 353
 contraintes informationnelles
 alias 171
 pseudonymes 169
 contraintes informationnelles sur les pseudonymes 170
 contrôle d'accès basé sur les intitulés (LBAC)
 fédéré 321
 contrôle d'accès basé sur les labels
 fédération 289
 sources de données prises en charge 353
 tables de requêtes matérialisées (MQT) 289
 CPU_RATIO (option de serveur)
 optimisation globale, incidence sur l' 275
 CREATE NICKNAME (instruction) 49
 CREATE TYPE MAPPING (instruction) 49, 51, 52, 53

D

DB2
 catalogue d'outils 181
 DB2 for Linux, UNIX et Windows
 mappages de types de données aval par défaut 439
 mappages de types de données inversés par défaut 453
 versions prises en charge 2
 DB2 for System i
 mappages de types de données aval par défaut 439
 mappages de types de données inversés par défaut 453
 prise en charge des objets LOB fédérés 217
 pseudonymes, objets valides pour 16
 versions prises en charge 2
 DB2 for z/OS
 prise en charge des objets LOB fédérés 217
 pseudonymes, objets valides pour 16
 versions prises en charge 2

DB2 for z/OS et OS/390
 mappages de types de données aval par défaut 439
 mappages de types de données inversés par défaut 453
 DB2_MAX_ASYNC_REQUESTS_PER_QUERY (option de serveur) 271
 DB2 pour Linux, UNIX et Windows
 prise en charge des objets LOB fédérés 217
 pseudonymes, objets valides pour 16
 DB2 pour VM et VSE
 mappages de types de données aval par défaut 439
 mappages de types de données inversés par défaut 453
 prise en charge des objets LOB fédérés 217
 pseudonymes, objets valides pour 16
 versions prises en charge 2
 DDL
 validation en deux phases fédérées 122
 DDL transparent
 longueurs de colonne LOB 104
 support des transactions pour 113
 Déclencheurs 156
 décrit 352
 définitions de serveur
 description 13
 dialecte SQL 163
 analyse de transfert de l'exécution sur la base de données, incidence sur l' 235
 description 12
 documentation
 accessible 475
 Documentation
 Accessible 471
 DRDA (Distributed Relational Database Architecture)
 configuration pour la validation en deux phases fédérées 128
 DUOW
 voir unités d'oeuvre réparties 134

E

encapsuleur XML 231
 encapsuleurs
 description 6
 Excel
 fonctions fédérées prises en charge 353
 options d'encapsuleur 369
 options de pseudonyme 369
 options de serveur 369
 exemples 146
 commande IMPORT 152
 validation en deux phases fédérées 118
 expressions de tables imbriquée
 tolérance aux erreurs 193

F

FEDERATED_ASYNC (paramètre de configuration) 271
 FEDERATED_ASYNCRONY (option de définition d'accès) 271
 Fichiers à plat
 Voir également fichiers structurés sous forme de tables 2
 Fichiers Excel
 pseudonymes, objets valides pour 16
 types de données, pris en charge 466
 versions prises en charge 2
 fichiers structurés sous forme de tables
 fonctions fédérées prises en charge 353
 options d'encapsuleur 413
 options de colonne 413
 options de pseudonyme 413
 options de serveur 413
 pseudonymes, objets valides pour 16
 types de données, pris en charge 466
 versions prises en charge 2
 fonction de mise à jour des statistiques de pseudonyme
 sources de données prises en charge 353
 Fonction Explain 252
 fonction XMLVALIDATE 189
 fonctions
 plug-in de mappage utilisateur 323
 fonctions définies par l'utilisateur (UDF) 18
 dans des applications de système fédéré 63
 support des transactions pour 113
 fonctions intégrées 18
 Fonctions SQL/XML 187

G

GROUP BY (opérateur)
 décisions d'évaluation de plan d'accès 245
 décisions d'optimisation de plan d'accès 282
 groupes de partition informatiques 260

H

heures, avec 24 dans la zone d'heures 146
 horodatages
 avec 24 dans la zone d'heures 146

I

identification et résolution des incidents
 validation en deux phases fédérées 132, 136
 importation de données
 sources de données prises en charge 353
 indicateurs d'état de santé fédérés
 sources de données prises en charge 353

- informations du catalogue distant 10
- Informix
 - configuration pour la validation en deux phases fédérées 129
 - fonctions fédérées prises en charge 353
 - mappages de types de données aval par défaut 439
 - mappages de types de données inversés par défaut 453
 - options d'encapsuleur 370
 - options de colonne 370
 - options de mappage utilisateur 370
 - options de serveur 370
 - prise en charge des objets LOB fédérés 217
 - pseudonymes, objets valides pour 16
 - versions prises en charge 2
- InfoSphere Data Architect 9
- Instruction CREATE FUNCTION (modèle ou dérivée) 61, 67
- Instruction CREATE FUNCTION MAPPING 61, 64
- Instruction CREATE INDEX 18, 71
- instruction CREATE PROCEDURE (dérivée)
 - procédures fédérées 90
- Instruction CREATE SERVER 5
- instruction DELETE 140
 - décisions d'évaluation de plan d'accès 245
- instruction INSERT 139, 140
 - décisions d'évaluation de plan d'accès 245
- Instruction SET SERVER OPTION
 - définition d'une option temporaire 13
- Instruction UPDATE 140
 - décisions d'évaluation de plan d'accès 245
- instructions SQL
 - alias 157
- intégrité référentielle 140
- interpréteur de commandes (CLP)
 - fonctions fédérées 7
- IO_RATIO (option de serveur)
 - optimisation globale, incidence sur l' 275
- isolement au niveau de l'instruction
 - sources de données prises en charge 213
 - systèmes fédérés 214
- isolement au niveau de la connexion
 - sources de données prises en charge 353
 - systèmes fédérés 215
- isolement de niveau d'instruction
 - sources de données prises en charge 353

J

- JDBC
 - fonctions fédérées prises en charge 353
 - options d'encapsuleur 375
 - options de colonne 375

- JDBC (*suite*)
 - options de mappage utilisateur 375
 - options de serveur 375
 - prise en charge des objets LOB fédérés 217
 - pseudonymes, objets valides pour 16
 - versions prises en charge 2
- jeux de caractères
 - description 19
- jointure d'ensembles de résultats
 - exemples de commande DB2FEDGENTF 98
 - syntaxe de commande DB2FEDGENTF 98
- jonctions
 - décisions d'optimisation de plan d'accès 282

L

- LBAC
 - fédération 289
 - tables de requêtes matérialisées (MQT) 289
- lecteurs d'écran 475
- Lecteurs d'écran 471
- LOB
 - sources de données prises en charge, lecture et écriture 353
 - sources de données prises en charge, lecture seule 353

M

- mappages de fonctions
 - analyse de transfert de l'exécution sur la base de données, incidence sur l' 235
 - création 64
 - description 18, 61
 - mappage vers les fonctions définies par l'utilisateur 64
 - mappages par défaut 61
 - options 435
- mappages de type aval
 - description 49
 - mappages par défaut 439
 - Unicode
 - Microsoft SQL Server 463
 - ODBC, sources de données 464
 - Sources de données JDBC 462
 - sources de données NET8 464
 - Sybase, sources de données 465
- mappages de type de données
 - analyse de transfert de l'exécution sur la base de données, incidence sur l' 235
 - aval 439
 - description 49
 - dans un système fédéré 48
 - date de création 47
 - description 17
 - inversé 453
 - description 49
 - pour un serveur particulier 53

- mappages de type de données (*suite*)
 - pour un type de source de données particulier 51
 - pour un type et une version de serveur particulier 52
 - syntaxe 49
 - transtypage 54
 - types de données non pris en charge 47
- mappages de type inversé
 - description 49
 - mappages par défaut 453
 - Unicode
 - Microsoft SQL Server, sources de données 463
 - ODBC, sources de données 465
 - Sources de données JDBC 462
 - sources de données NET8 464
 - Sybase, sources de données 466
- mappages de types de données
 - cas exigeant de nouveaux mappages 49
 - non relationnelles 49
 - pour un objet de source de données particulier 56
- mappages de types de données aval par défaut
 - exemple pour Informix 451
 - exemple pour Microsoft SQL Server 452
 - Exemple pour Oracle 452
 - exemple pour Sybase 452
 - Exemple pour Teradata 453
 - exemples 451
 - Microsoft SQL Server, sources de données 445, 458
 - ODBC, sources de données 447
 - sources de données d'une base de données DB2 pour Linux, UNIX et Windows 439
 - sources de données DB2 for System i 440
 - sources de données DB2 for z/OS 442
 - sources de données DB2 pour VM et VSE 441
 - sources de données Informix 442, 457
 - Sources de données JDBC 444
 - sources de données Oracle NET8 448
 - Sybase, sources de données 448
 - Teradata, sources de données 450, 461
- mappages de types de données inversés par défaut
 - ODBC, sources de données 459
 - sources de données d'une base de données DB2 pour Linux, UNIX et Windows 454
 - sources de données DB2 for System i 455
 - sources de données DB2 for z/OS 456
 - sources de données DB2 pour VM et VSE 455
 - Sources de données JDBC 457
 - sources de données Oracle NET8 460

mappages de types de données inversés
par défaut (*suite*)

Sybase, sources de données 460

mappages utilisateur

description 14

plug-ins échantillons 323

référentiels externes 323

stockage 14, 323

marques 479

membre d'un ensemble d'abonnements,
tables de mémoire cache 296

Microsoft Excel

Voir Fichiers Excel 2

Microsoft SQL Server

configuration pour la validation en
deux phases fédérées 130

fonctions fédérées prises en

charge 353

mappages de types de données aval
par défaut 439

mappages de types de données aval
par défaut Unicode 463

mappages de types de données

inversés par défaut 453

options d'encapsuleur 382

options de colonne 382

options de mappage utilisateur 382

options de serveur 382

prise en charge des objets LOB
fédérés 217

pseudonymes, objets valides pour 16

versions prises en charge 2

Microsoft SQL Server, sources de données

mappages de types de données aval
par défaut 445, 458

mappages de types de données

inversés par défaut Unicode 463

procédures fédérées 83

mise à jour multisite, fédérée

voir validation en deux phases

fédérées 115

mise en mémoire cache 207, 287, 288

alias 321

misés à jour

à distance 113

autorisation 139

description 113

intégrité référentielle 140

locales 113

restrictions 140

vers des objets LOB 141

misés à jour du statut

alias 181

misés à jour éloignées 113

misés à jour locales 113

modèles de fonction

description 67

transfert de l'exécution sur la base de
données du prédicat 249

modification 24

types de données LONG 59

MQT (tables de requêtes matérialisées)

fédéré

présentation 287

restrictions sur les pseudonymes 289

Oracle Label Security (OLS) 289

sur les pseudonymes 242

N

nickname

méthodes d'extraction 176

noms d'index 162

noms de colonnes 162

niveaux d'isolement

systèmes fédérés 213

Notices légales 477

NUMERIC_STRING (option de colonne)

possibilités de transfert de l'exécution

sur la base de données, incidence

sur les 242

O

objets de source de données 160

description 15

types d'objet valides 16

objets éloignés, pseudonymes 155

objets locaux 155

ODBC

fonctions fédérées prises en

charge 353

mappages de types de données aval

par défaut 439

options d'encapsuleur 388

options de colonne 388

options de mappage utilisateur 388

options de serveur 388

prise en charge des objets LOB

fédérés 217

pseudonymes, objets valides pour 16

versions prises en charge 2

ODBC, sources de données

mappages de types de données aval

par défaut 447

mappages de types de données aval

par défaut Unicode 464

mappages de types de données

inversés par défaut 459

mappages de types de données

inversés par défaut Unicode 465

OLE DB

versions prises en charge 2

OLS (Oracle Label Security)

restrictions sur les pseudonymes

tables de requêtes matérialisées

(MQT) 289

opérateurs ensemblistes

décisions d'évaluation de plan

d'accès 245

opérations d'écriture

voir misés à jour 113

opérations de validation en une phase

définies 111

opérations heuristiques

résolution des transactions en attente

de validation

systèmes fédérés 133

Optim Data Studio 9

optimisation

caractéristiques de serveur, incidence

sur les 275

options de colonne de

pseudonyme 242

options de serveur 235

optimisation (*suite*)

requêtes asynchrones 272

séquences de classement 235

spécifications d'index 275

statistiques de catalogue 275

Tables de requêtes matérialisées 242

traitement de requête 232

validation en deux phases

fédérées 137

amélioration des

performances 138

Voir aussi - Performances 232

optimisation de l'état asynchrone

sources de données prises en

charge 353

optimisation de requête

description 11

Optimisation globale 249

caractéristiques de serveur, incidence

sur les 275

description 275

optimiseur

description 11

option, WITH HOLD 156

option DB2_UM_PLUGIN 334

option DB2_UM_PLUGIN_LANG 334

option de mappage de fonctions

DISABLE

paramètres valides 435

option de mappage de fonctions

REMOTE_NAME

paramètres valides 435

option de serveur

COLLATING_SEQUENCE

exemple 19

optimisation globale, incidence sur

l' 275

possibilités de transfert de l'exécution

sur la base de données, incidence

sur les 235

option de serveur

DB2_MAXIMAL_PUSHDOWN

décisions de l'analyse de transfert de

l'exécution sur la base de

données 244

possibilités de transfert de l'exécution

sur la base de données, incidence

sur les 235

option de serveur

IUD_APP_SVPT_ENFORCE

exemples 141

option de serveur

VARCHAR_NO_TRAILING_BLANKS

possibilités de transfert de l'exécution

sur la base de données, incidence

sur les 235

options d'encapsuleur

Base de données DB2 361

BioRS 357

Excel 369

fichiers structurés sous forme de

tables 413

Informix 370

JDBC 375

Microsoft SQL Server 382

ODBC 388

Oracle 393

- options d'encapsuleur (*suite*)
 - Script 399
 - services Web 416
 - Sybase 404
 - Teradata 409
 - XML 424
 - options de colonne
 - analyse de transfert de l'exécution sur la base de données, incidence sur l' 242
 - Base de données DB2 361
 - BioRS 357
 - description 16
 - fichiers structurés sous forme de tables 413
 - Informix 370
 - JDBC 375
 - Microsoft SQL Server 382
 - ODBC 388
 - Oracle 393
 - Script 399
 - services Web 416
 - Sybase 404
 - Teradata 409
 - XML 424
 - options de colonne de pseudonyme
 - description 16
 - options de mappage utilisateur
 - Base de données DB2 361
 - BioRS 357
 - Informix 370
 - JDBC 375
 - Microsoft SQL Server 382
 - ODBC 388
 - Oracle 393
 - Script 399
 - services Web 416
 - Sybase 404
 - Teradata 409
 - XML 424
 - options de pseudonyme
 - Base de données DB2 361
 - BioRS 357
 - fichiers structurés sous forme de tables 413
 - Script 399
 - services Web 416
 - XML 424
 - options de serveur
 - analyse de transfert de l'exécution sur la base de données, incidence sur l' 235
 - Base de données DB2 361
 - BioRS 357
 - description 13
 - état asynchrone 271
 - Excel 369
 - fichiers structurés sous forme de tables 413
 - Informix 370
 - JDBC 375
 - Microsoft SQL Server 382
 - ODBC 388
 - optimisation globale, incidence sur l' 275
 - Oracle 393
 - Script 399
 - options de serveur (*suite*)
 - services Web 416
 - Sybase 404
 - temporaire 13
 - Teradata 409
 - XML 424
 - Oracle 352
 - chaînes vides 146
 - configuration pour la validation en deux phases fédérées 129
 - fonctions fédérées prises en charge 353
 - mappages de types de données aval par défaut 439
 - mappages de types de données inversés par défaut 453
 - options d'encapsuleur 393
 - options de colonne 393
 - options de mappage utilisateur 393
 - options de serveur 393
 - prise en charge des objets LOB fédérés 217
 - pseudonymes, objets valides pour 16
 - résolution des incidents de validation en deux phases fédérées 136
 - Oracle, sources de données
 - procédures fédérées 84
 - procédures multiréférencées 86
 - sécurité 351
 - Oracle Label Security
 - description 351
 - Oracle Label Security (OLS)
 - restrictions sur les pseudonymes tables de requêtes matérialisées (MQT) 289
 - ORDER BY (opérateur)
 - décisions d'évaluation de plan d'accès 245
 - outil db2exfmt
 - affichage des plans d'accès 244, 281
 - outil db2expln
 - affichage des plans d'accès 244, 281
 - outil dynexpln
 - affichage des plans d'accès 244, 281
 - outils Explain 263
- P**
- pages de codes
 - description 19
 - parallélisme 251, 260, 262
 - fédéré 251, 254
 - parallélisme interpartition 251
 - fédéré 254, 258, 260, 262
 - parallélisme intrapartition 251
 - fédéré 251
 - plans d'accès fédérés 252
 - parallélisme mixte
 - sources de données fédérées
 - plan d'accès 263
 - présentation 251
 - traitement des données 262
 - pas-système
 - description 13
 - Prise en charge des objets LOB 218
 - restrictions 13
 - support des transactions pour 113
 - performances 169, 234, 244, 260, 262
 - cadence de l'unité centrale 275
 - débit de communication 275
 - Différences SQL 235
 - fédéré 171, 174, 180, 182, 231
 - indications de plan éloigné 275
 - séquence de classement 275
 - séquences de classement 235
 - traitement de requête
 - asynchrone 265
 - validation en deux phases fédérées 137
 - amélioration 138
 - vitesse des E-S 275
 - Voir aussi - Optimisation 232
 - plan d'accès 263
 - PLAN_HINTS (option de serveur)
 - optimisation globale, incidence sur l' 275
 - planification
 - validation en deux phases fédérées 116
 - plans d'accès 258
 - affichage 244, 281
 - décisions d'évaluation 245
 - décisions d'optimisation 282
 - description 11
 - optimisation de l'état asynchrone 267
 - performances 282
 - plug-ins de mappage utilisateur
 - accès serveurs fédérés 334
 - langage C
 - désignation 334
 - génération 333
 - langage de programmation C
 - déclaration de fonctions 331
 - déclarations de fonction 330
 - déploiement 334
 - développement 330
 - fichier d'en-tête 326
 - fonction FSUMconnect 329
 - fonction FSUMdisconnect 330
 - fonction FSUMfetchUM 329
 - fonction FSUMPluginInit 328
 - fonction FSUMPluginTerm 330
 - fonctions 323
 - mise à jour 335
 - plates-formes prises en charge 325
 - plug-in d'exemple 335
 - pointeurs de fonction 331
 - présentation 323
 - restrictions 325
 - test 333
 - traitement d'erreurs 332
 - langage de programmation Java
 - architecture 336
 - classes 338, 339, 340, 341
 - compilation de fichiers 346
 - configuration de l'accès 349
 - déploiement 349
 - développement 343, 344
 - fichier de configuration 347
 - fichiers d'exemple 342, 344, 345
 - test 348
 - option DB2_UM_PLUGIN 334

- plug-ins de mappage utilisateur (*suite*)
 - option
 - DB2_UM_PLUGIN_LANG 334
- points de sauvegarde
 - API de source de données 141
 - dans la fédération 148
- points de sauvegarde d'une application 148
 - sources de données prises en charge 353
- prédicats
 - avec des modèles de fonction 249
 - décisions d'évaluation de plan d'accès 245
- présentation
 - validation en deux phases fédérées 115
- Procédure mémorisée
 - SYSPROC.NNSTAT 182
- procédures
 - fédéré 79
 - fédérées, création 90
 - fédérées, identification et résolution des incidents 99
 - fédérées, jointure d'ensembles de résultats 95
 - fédérées, suppression 94
- procédures fédérées 19
 - appel 93
 - création 90
 - DB2 81
 - identification et résolution des incidents 99
 - instruction CREATE PROCEDURE (dérivée) 90
 - jointure d'ensembles de résultats 95
 - Microsoft SQL Server 83
 - Oracle 84
 - paramètres 93
 - présentation 79
 - privilèges, octroi 91
 - privilèges, révocation 91
 - sources de données prises en charge 353
 - support de source de données 81
 - suppression 94
 - Sybase 88
 - vues catalogue 93
- procédures multiréférencées
 - procédures fédérées 86
- procédures stockées
 - statistiques des pseudonymes 182
- programmes d'application 63
- Proxy HTTP
 - sources de données prises en charge 353
- Proxy SOCKS
 - sources de données prises en charge 353
- pseudonymes
 - contraintes informationnelles 169
 - modification
 - type de données local, exemple 56

R

- référentiel de mappage utilisateur
 - sources de données prises en charge 353
- Registre spécial CURRENT FEDERATED ASYNCHRONY 271
- Registre spécial CURRENT IMPLICIT XMLPARSE OPTION 191
- règles
 - sémantique de l'attribution fédérée 144
- requêtes
 - fragments 11
 - routage, tables de mémoire cache 295
 - tables de mémoire, routage 295
 - traitement asynchrone 265
- résolution des incidents de validation en deux phases fédérées 136

S

- Script
 - fonctions fédérées prises en charge 353
 - options d'encapsuleur 399
 - options de colonne 399
 - options de mappage utilisateur 399
 - options de pseudonyme 399
 - options de serveur 399
 - versions prises en charge 2
- Secure Socket Layer (SSL)
 - sources de données prises en charge 353
- sécurité 352
- Sécurité
 - Oracle 351
- sémantiques d'attribution fédérées
 - exemples 146
- séquences de classement
 - description 19
 - planification 19
 - présentation 235
- serveur fédéré 5
 - description 5
 - sémantique VARCHAR2 236
- serveur LDAP
 - mappages utilisateur 323
- serveurs proxy
 - décrit 299
- services Web
 - fonctions fédérées prises en charge 353
 - options d'encapsuleur 416
 - options de colonne 416
 - options de mappage utilisateur 416
 - options de pseudonyme 416
 - options de serveur 416
 - types de données, pris en charge 466
- sessions passe-système 163
 - sources de données prises en charge 353
- source de données
 - sémantique VARCHAR2 236

- source de données JDBC
 - mappages de types de données aval par défaut 444
 - mappages de types de données aval par défaut Unicode 462
 - mappages de types de données inversés par défaut 457
 - mappages de types de données inversés par défaut Unicode 462
- sources de données 6, 11
 - création de pseudonymes 161
 - débit de communication et performances 275
 - description 5
 - exigences pour la validation en deux phases fédérées 127
 - fonctions prises en charge 353
 - indications de plan éloigné et performances 275
 - options 357
 - séquence de classement et performances 275
 - types de serveur valides 437
 - vitesse des E-S et performances 275
 - vitesse du processeur et performances 275
- sources de données d'une base de données DB2 pour Linux, UNIX et Windows
 - mappages de types de données aval par défaut 439
 - mappages de types de données inversés par défaut 454
- sources de données DB2
 - procédures fédérées 81
- Sources de données DB2 for System i
 - mappages de types de données aval par défaut 440
 - mappages de types de données inversés par défaut 455
- sources de données DB2 for z/OS
 - mappages de types de données aval par défaut 442
 - mappages de types de données inversés par défaut 456
- sources de données DB2 pour VM et VSE
 - mappages de types de données aval par défaut 441
 - mappages de types de données inversés par défaut 455
- sources de données Informix
 - mappages de types de données aval par défaut 442, 457
- sources de données NET8
 - mappages de types de données aval par défaut Unicode 464
 - mappages de types de données inversés par défaut Unicode 464
- sources de données non relationnelles
 - spécification des mappages de type de données 17
 - types de données pris en charge 466
- Sources de données Oracle NET8
 - mappages de types de données aval par défaut 448
 - mappages de types de données inversés par défaut 460

- Sources de données Sybase
 - mappages de types de données aval par défaut 448
 - mappages de types de données aval par défaut Unicode 465
 - mappages de types de données inversés par défaut 460
 - mappages de types de données inversés par défaut Unicode 466
- spécifications d'index
 - description 18
 - fédéré 71
- SQL compiler
 - dans un système fédéré 11
 - graphique de flux de l'analyse de requête 232
- SQL Explain
 - affichage des plans d'accès 244, 281
- statistiques
 - collecte automatique 184
 - HIGH2KEY 180
 - LOW2KEY 180
 - méthodes d'extraction 176
 - nickname 174, 176, 180
 - plusieurs pseudonymes 178
 - pseudonyme unique 179
- statistiques fédérées
 - mise à jour 174
- statistiques HIGH2KEY 180
- statistiques LOW2KEY 180
- surveillance d'instantané 200
 - fragments de requête fédérée 202
 - pseudonymes et serveurs 199
 - pseudonymes et serveurs fédérés 197
- Sybase 136
 - configuration pour la validation en deux phases fédérées 131
 - fonctions fédérées prises en charge 353
 - mappages de types de données aval par défaut 439
 - mappages de types de données inversés par défaut 453
 - options d'encapsuleur 404
 - options de colonne 404
 - options de mappage utilisateur 404
 - options de serveur 404
 - prise en charge des objets LOB fédérés 217
 - pseudonymes, objets valides pour 16
 - versions prises en charge 2
- Sybase, sources de données
 - procédures fédérées 88
- syntaxe WITH HOLD 156
- système de gestion de base de données réparties 1
- systèmes fédérés
 - isolement au niveau de l'instruction 214
 - isolement au niveau de la connexion 215
 - niveaux d'isolement 213
 - présentation 1

T

- table SYSPROC.FED_STATS 181
- tables de mémoire cache 289
 - activation 296
 - affichage des paramètres 294
 - assistant Table de mémoire cache 293
 - conditions requises 293
 - consignation des archives 293
 - création 293
 - désactivation 296
 - description 291
 - modification 294
 - plannings, répllication 291
 - répllication 291
 - routage des requêtes 295
 - sources de données 293
 - sources de données prises en charge 353
 - suppression 297
 - Tables de requêtes matérialisées 294
- tables de requêtes matérialisées
 - tables de mémoire cache 291
- Tables de requêtes matérialisées
 - ajout à des tables de mémoire cache 294
 - sources de données prises en charge 353
 - suppression, des tables de mémoire cache 297
- tables de requêtes matérialisées (MQT) 207, 288
 - fédéré
 - présentation 287
 - restrictions sur les pseudonymes 289
 - sur les pseudonymes 242
- Teradata
 - fonctions fédérées prises en charge 353
 - mappages de types de données aval par défaut 439
 - mappages de types de données inversés par défaut 453
 - options d'encapsuleur 409
 - options de colonne 409
 - options de mappage utilisateur 409
 - options de serveur 409
 - prise en charge des objets LOB fédérés 217
 - pseudonymes, objets valides pour 16
- Teradata, sources de données
 - mappages de types de données aval par défaut 450, 461
- TIMESTAMP 55
- tolérance aux erreurs
 - activation 194
 - description 193
 - exemple 195
 - restrictions 196
 - sources de données prises en charge 353
 - support de source de données 196
- traitement asynchrone
 - activation 271
 - description 265
 - exemples 265
 - optimisation 273
 - restrictions 273

- transactions
 - mises à jour 113
 - présentation 111
- transactions en attente de validation
 - récupération 132
 - résolution
 - systèmes fédérés 133
 - resynchronisation pour les systèmes fédérés 132
 - traçage des unités d'oeuvre réparties 134
- transtypage 54
- tri 19
- type de données DATALINK
 - non pris en charge 17
- types de données 104
 - analyse de transfert de l'exécution sur la base de données, incidence sur l' 242
 - non pris en charge 17
 - pour les sources de données non relationnelles 466
 - sémantique VARCHAR2 236
 - TIMESTAMP 55
- types de données d'objet LOB
 - considérations de performances 219
 - opérations de mise à jour 141
 - releveurs de coordonnées 218
 - restrictions 218
- types de données LOB (large object)
 - opérations de mise à jour 141
 - releveurs de coordonnées 218
 - restrictions 218
- types de données LONG 59
- types de serveur
 - types fédérés valides 437
- types UDT
 - types de données non pris en charge 17

U

- Unicode
 - sources de données prises en charge 353
- unités d'oeuvre réparties
 - traçage dans les sources de données 134

V

- validation en deux phases
 - opérations 111
 - sources de données prises en charge 353
- validation en deux phases fédérées
 - activation 125
 - amélioration des performances 138
 - architecture 116
 - atomicité 122
 - configuration 125
 - configuration requise pour la source de données 127
 - configurations 122
 - DDL 122
 - DDL transparent 122

- validation en deux phases fédérées (*suite*)
 - validation en deux phases fédérées 122
 - exemples 118
 - identification et résolution des incidents 132
 - opérations admises 122
 - performances 137
 - planification 116
 - présentation 115
- validation en deux phases pour les transactions fédérées
 - voir validation en deux phases fédérées 115
- VARCHAR_NO_TRAILING_BLANKS
 - (option de colonne)
 - possibilités de transfert de l'exécution sur la base de données, incidence sur les 242
- VARCHAR2
 - sémantique 236
- Visual Explain
 - affichage des plans d'accès 244, 281
- vues catalogues SYSCAT 64
- vues de catalogue SYSCAT 431
- vues de catalogue SYSSTAT 431
- vues fédérées
 - accès aux données 163
 - création 165

X

- XML
 - documents
 - décomposition 190
 - validation 189
 - options d'encapsuleur 424
 - options de colonne 424
 - options de mappage utilisateur 424
 - options de pseudonyme 424
 - options de serveur 424
 - pseudonymes, objets valides pour 16
 - référentiel de schémas 190
 - schémas, enregistrement 188
 - type de données
 - restrictions 191
 - support 187
 - types de données, pris en charge 466
 - versions prises en charge 2

Z

- zone d'heures 146



SC11-6629-00



Spine information:

IBM InfoSphere Federation Server

Version 9.7

Guide d'administration pour des systèmes fédérés

