



Guía de administración para sistemas federados



Guía de administración para sistemas federados

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información de la sección "Avisos" en la página 467.

Contenido

Capítulo 1. Sistemas federados. 1

Orígenes de datos soportados.	2
El servidor federado.	5
Qué es un origen de datos.	6
La base de datos federada.	6
Derivadores y módulos de derivador.	7
Cómo interactuar con un sistema federado.	8
Centro de control de DB2.	9
Programas de aplicación.	9
Herramientas de la familia DB2.	10
IBM Optim Data Studio.	10
Proveedores de servicios Web.	10
Catálogo del sistema de bases de datos federadas.	11
Compilador de SQL.	11
Optimizador de consultas.	12
Compensación.	12
Sesiones de paso a través.	13
Definiciones de servidor y opciones de servidor.	14
Correlaciones de usuario.	15
Apodos y objetos de origen de datos.	15
Objetos de origen de datos válidos.	16
Opciones de columna de apodo.	17
Correlaciones de tipos de datos.	18
Correlaciones de funciones.	18
Especificaciones de índice.	19
Procedimientos almacenados federados.	19
Secuencias de clasificación.	19
Cómo determinan las secuencias de clasificación los órdenes de clasificación.	20
Establecimiento de la secuencia de clasificación local para optimizar consultas.	21

Capítulo 2. Modificación de configuraciones de orígenes de datos. 23

Modificación de un derivador (Centro de control de DB2).	23
Modificación de un derivador - ejemplos.	24
Modificación de un derivador (línea de mandatos de DB2).	24
Modificación de definiciones de servidor y opciones de servidor.	25
Restricciones en la modificación de definiciones de servidor.	26
Modificación de la versión de origen de datos en una definición de servidor (Centro de control de DB2).	26
Modificación de la versión de origen de datos en una definición de servidor (línea de mandatos de DB2).	27
Modificación de todas las definiciones de servidor para un tipo de origen de datos específico.	28
Utilización de opciones de servidor en definiciones de servidor (Centro de control de DB2).	28

Modificación temporal de opciones de servidor para orígenes de datos relacionales.	29
Jerarquía de los valores de opciones de servidor.	29
Utilización de opciones de servidor en definiciones de servidor (línea de mandatos de DB2).	30
Modificación de una correlación de usuario (Centro de control de DB2).	30
Modificación de una correlación de usuario (línea de mandatos de DB2).	31
Modificación de un apodo (Centro de control de DB2).	33
Restricciones en la modificación de apodos.	34
Modificación de nombres de columnas de apodo (Centro de control de DB2).	35
Modificación de nombres de columnas de apodo (línea de mandatos de DB2).	36
Modificación de opciones de apodo (Centro de control de DB2).	37
Modificación de opciones de apodo (línea de mandatos de DB2).	38
Modificación de opciones de columnas de apodo (Centro de control de DB2).	38
Modificación de opciones de columnas de apodo (línea de mandatos de DB2).	39
Modificación de un apodo (línea de mandatos de DB2).	41
Descarte de un derivador.	42
Descarte de una definición de servidor.	43
Descarte de una correlación de usuario.	44
Descarte de un apodo.	44

Capítulo 3. Correlaciones de tipos de datos en un sistema federado. 47

Correlaciones de tipos de datos y el catálogo global de bases de datos federadas.	47
Cuándo se deben crear correlaciones de tipos de datos alternativas.	48
Correlaciones de tipos de datos para orígenes de datos no relacionales.	49
Correlaciones de tipos de datos directas e inversas.	49
Creación de correlaciones de tipos de datos.	50
Creación de una correlación de tipos para un tipo de origen de datos - ejemplo.	51
Creación de una correlación de tipos para un tipo de origen de datos y versión - ejemplo.	52
Creación de una correlación de tipos para todos los objetos de origen de datos en un servidor - ejemplo.	53
Conversión entre tipos de datos.	54
Soporte para tipos de datos TIMESTAMP.	55
Modificación de un tipo local para un objeto de origen de datos (Centro de control de DB2).	55
Modificación de un tipo local para un objeto de origen de datos - ejemplos.	56
Modificación de un tipo local para un objeto de origen de datos (línea de mandatos de DB2).	57

Modificación de tipos de datos LONG por tipos de datos VARCHAR	58
--	----

Capítulo 4. Correlaciones de funciones y funciones definidas por el usuario . . 61

Correlaciones de funciones en un sistema federado	61
Funcionamiento de las correlaciones de funciones en un sistema federado	61
¿Por qué son importantes las correlaciones de funciones?	62
Cuándo crear correlaciones de funciones.	62
Funciones definidas por el usuario en aplicaciones	63
Requisitos para correlacionar funciones definidas por el usuario	63
Crear correlaciones de funciones	64
Especificación de nombres de función en la sentencia CREATE FUNCTION MAPPING	64
Creación de una correlación de funciones para un tipo de origen de datos específico	65
Creación de una correlación de funciones para un tipo de origen de datos y una versión específicos.	66
Creación de una correlación de funciones para todos los objetos de origen de datos en un servidor específico	66
Plantillas de función	67
Creación de plantillas de funciones	67
Inhabilitación de una correlación de funciones predeterminada	69
Descarte de una correlación de funciones	69

Capítulo 5. Creación de especificaciones de índice 71

Especificaciones de índice en un sistema federado	71
Creación de especificaciones de índice para objetos de origen de datos	72
Creación de especificaciones de índice sobre tablas que adquieren índices nuevos	73
Creación de especificaciones de índice en vistas	74
Creación de especificaciones de índice sobre sinónimos de Informix.	75

Capítulo 6. Desarrollo de procedimientos federados 79

Procedimientos federados	79
Orígenes de datos y procedimientos federados	81
Procedimientos federados para orígenes de datos DB2	81
Procedimientos federados y Microsoft SQL Server	82
Procedimientos federados y Oracle	84
Procedimientos federados y Sybase	88
Creación de procedimientos federados	90
Cómo otorgar o revocar autorizaciones para llamar a procedimientos federados	91
Visualización de información de parámetros y llamada a procedimientos federados	93
Modificación o eliminación de procedimientos federados	94
Unión de conjuntos de resultados de procedimientos federados	95
Mandato DB2FEDGENTF.	98

Resolución de problemas de procedimientos federados	99
---	----

Capítulo 7. Creación y modificación de tablas remotas utilizando DDL transparente 105

¿Qué es el DDL transparente?	105
Columnas LOB remotas y DDL transparente	106
Creación de tablas remotas y DDL transparente	106
Creación de tablas remotas nuevas utilizando DDL transparente	107
Creación de tablas remotas nuevas utilizando DDL transparente - ejemplos	108
Modificación de tablas remotas utilizando DDL transparente	110
Descarte de tablas remotas utilizando DDL transparente	111

Capítulo 8. Gestión de transacciones en un sistema federado 113

Descripción del soporte de transacción de un sistema federado	113
Qué es una actualización en un sistema federado	114
Qué es una transacción de actualización en una sesión de paso a través	115
Orígenes de datos con confirmación automática de las sentencias de DDL	116
Funciones definidas por el usuario que son enviadas al origen de datos para su proceso	116

Capítulo 9. Realización de transacciones de confirmación en dos fases 117

Confirmación en dos fases para transacciones federadas.	117
Planificación para la confirmación federada en dos fases	117
Arquitectura federada para la confirmación en dos fases	118
Confirmación en dos fases para transacciones federadas - ejemplos	120
Proceso de las transacciones federadas con confirmación en dos fases	124
Habilitación de la confirmación en dos fases para las transacciones federadas	128
Requisitos y configuración del origen de datos para las transacciones de confirmación federada en dos fases	129
Configuración de orígenes de datos DRDA	130
Configuración de orígenes de datos Oracle	131
Configuración de orígenes de datos Informix	132
Configuración de orígenes de datos de Microsoft SQL Server.	133
Configuración de orígenes de datos Sybase	133
Recuperación de problemas de confirmación federada en dos fases.	135
Resincronización para sistemas federados	135
Recuperación manual de transacciones dudosas	136

Rastreo de los estados de una transacción de unidad de trabajo distribuida entre los orígenes de datos	137
Resolución de problemas de la confirmación federada en dos fases.	138
Rendimiento de la confirmación federada en dos fases	139
Mejora del rendimiento de la confirmación federada en dos fases.	140

Capítulo 10. Insertar, actualizar y suprimir datos en un sistema federado 143

Privilegios de autorización para sentencias INSERT, UPDATE y DELETE	143
Restricciones sobre INSERT, UPDATE y DELETE en un sistema federado	144
Orígenes de datos no soportados	144
Integridad referencial en un sistema federado	144
Sentencias INSERT, UPDATE y DELETE y objetos grandes (LOB)	145
Preservación de la atomicidad de las sentencias en un sistema federado	145
Modificación de datos en un sistema federado	146
Inserción de datos en objetos de origen de datos	146
Actualización de datos en objetos de origen de datos	146
Supresión de datos de objetos de origen de datos	147
Semántica de asignación en un sistema federado	148
Semántica de asignación en un sistema federado - ejemplos	150
Restricciones de orígenes de datos para los valores de tipos de datos	150
Actualizaciones federadas con puntos de rescate de aplicación	152
Actualizaciones federadas con puntos de rescate de aplicación - ejemplos	152
Restricciones sobre las actualizaciones federadas con puntos de rescate de aplicación	153

Capítulo 11. Importación y exportación de datos para apodos . . . 155

Restricciones al importar datos en apodos	155
Utilización del mandato IMPORT con apodos - ejemplos	156
Restricciones para exportar datos utilizando apodos	156

Capítulo 12. Trabajar con apodos. . . 157

Apodos en un sistema federado	157
Cursores declarados WITH HOLD	157
Desencadenantes	157
Acceso a datos mediante apodos	158
Sentencias SQL que pueden utilizarse con apodos	158
Acceso a nuevos objetos de origen de datos	162
Creación de apodos para orígenes de datos relacionales y no relacionales	163
Nombres de índice y de columnas de apodos	164

Acceso a orígenes de datos mediante sesiones de paso a través	164
Acceso a datos heterogéneos mediante vistas federadas.	165
Creación de vistas federadas - ejemplos	166
Creación de un apodo para un apodo	167
Selección de datos en un sistema federado	167
Selección de datos en un sistema federado - ejemplos	168
Restricciones informativas para apodos.	170
Especificación de restricciones informativas para apodos (Centro de control de DB2)	171
Especificación de restricciones informativas para apodos (línea de mandatos de DB2)	171
Especificación de restricciones informativas para apodos - ejemplos	172
Actualización de estadísticas para apodos	175
Recurso de actualización de estadísticas de apodo - Visión general	175
Métodos de obtención de estadísticas de apodos	177
Obtención de estadísticas de apodo	177
Restricciones en las estadísticas de HIGH2KEY y LOW2KEY	181
Creación de un catálogo de herramientas de DB2	181
Visualización del estado de las actualizaciones hechas en estadísticas de apodo (Centro de control de DB2).	182
Visualización del estado de las actualizaciones realizadas en estadísticas de apodo (línea de mandatos de DB2).	182
Procedimiento almacenado SYSPROC.NNSTAT	183
Actualización automática de estadísticas de apodos	185

Capítulo 13. Trabajar con datos XML remotos. 187

Creación de un apodo para datos XML remotos - ejemplos	187
Manipulación de datos XML - ejemplos	187
Validación de documentos XML contra esquemas XML	188
Registro de esquemas XML.	188
Validación de documentos XML	189
Descomposición de documentos XML con esquemas XML anotados en apodos	190
Proceso federado de datos XML remotos	190
Análisis federado de datos XML remotos	190
Problemas de página de códigos para datos XML remotos	191
Restricciones sobre tipos de datos XML remotos	191

Capítulo 14. Tolerancia a errores en expresiones de tabla anidadas 193

Especificación de expresiones de tabla anidadas para la tolerancia a errores	194
Expresiones de tabla anidadas para la tolerancia a errores - ejemplo	195
Soporte de orígenes de datos para expresiones de tabla anidadas para la tolerancia a errores.	195

Restricciones respecto a expresiones de tabla
anidadas para la tolerancia a errores. 196

Capítulo 15. Supervisión de apodos y servidores federados 197

Indicadores de salud para servidores y apodos
federados. 197
Activación de los indicadores de salud federados 198
Supervisión de estado de los servidores y los
apodos federados 198
 Supervisión de estado de los servidores y los
 apodos federados - ejemplo 199
Supervisión de instantánea de sistemas federados -
Visión general 200
 Supervisión de consultas federadas 200
 Supervisión de instantánea de consultas
 federadas - Visión general 202
Elementos del supervisor de sistemas de base de
datos federada 203

Capítulo 16. Cómo interactúan las aplicaciones cliente con los orígenes de datos 205

Capítulo 17. Referencia a objetos de origen de datos por medio de apodos en sentencias SQL 207

Apodos en sentencias DDL. 207
Impacto de las estadísticas del origen de datos en
las aplicaciones. 208
Definición de opciones de columna en apodos . . . 209
 Cómo establecer la opción de columna
 NUMERIC_STRING 209
 Cómo establecer la opción de columna
 VARCHAR_NO_TRAILING_BLANKS 210

Capítulo 18. Creación y utilización de vistas federadas 211

Creación de vistas federadas - ejemplos 211

Capítulo 19. Mantener la integridad de los datos con niveles de aislamiento . 213

Aislamiento de nivel de sentencia en un sistema
federado 214
Aislamiento de nivel de conexión en un sistema
federado 215

Capítulo 20. Soporte de LOB federados 217

Localizadores de LOB 218
Restricciones sobre los LOB 218
Consideraciones de rendimiento para proceso de
LOB 219

Capítulo 21. Peticiones distribuidas para consultar orígenes de datos. . . 221

Peticiones distribuidas para consultar orígenes de
datos - ejemplos 221
Optimización de peticiones distribuidas con
opciones de servidor 222
Cancelar una consulta federada 223

Capítulo 22. Consulta directa de orígenes de datos con paso a través . 225

Consideraciones y restricciones de paso a través
federado 225
Sesiones de paso a través para orígenes de datos
Oracle. 227

Capítulo 23. Ajuste del proceso de consultas 229

Publicaciones sobre el rendimiento federado . . . 229
Análisis de consultas 230
Análisis de envío 232
Características de servidor que afectan a
oportunidades de envío 233
 Diferencias de SQL 233
 Secuencia de clasificación 236
 Opciones de servidor federado 238
 Factores de correlación de tipos y funciones . . 239
Características de apodo que afectan a
oportunidades de envío 240
 Tipo de datos locales de una columna de apodo 240
 Opciones de columna federada 240
Características de consulta que afectan a
oportunidades de envío 241
Análisis del lugar donde se evalúa una consulta 241
 Análisis del lugar donde se evalúa una consulta
 con la opción de servidor
 DB2_MAXIMAL_PUSHDOWN 242
Interpretación de decisiones de evaluación de
planes de acceso 243
 Por qué no se evalúa este predicado de forma
 remota 243
 Por qué el operador GROUP BY no se evalúa de
 forma remota 243
 Por qué no se evalúa el operador SET de forma
 remota 244
 Por qué no se evalúa la operación ORDER BY
 de forma remota 244
 Por qué no se evalúa completamente una
 sentencia INSERT con fullselect de forma
 remota 244
 Por qué no se evalúa completamente una
 sentencia remota INSERT con cláusula VALUES
 de forma remota 245
 Por qué una sentencia UPDATE remota y
 buscada no se evalúa completamente de forma
 remota 245
 Por qué una sentencia UPDATE posicionada no
 se evalúa completamente de forma remota . . 245
 Por qué una sentencia DELETE remota y
 buscada no se evalúa completamente de forma
 remota 246
Actualizaciones y personalización de orígenes de
datos 246

Envío de predicados con plantillas de función . . . 246

Capítulo 24. Paralelismo con consultas que hacen referencia a apodos 249

Paralelismo intrapartición con consultas que hacen referencia a apodos	249
Habilitación del paralelismo intrapartición con consultas que hacen referencia a apodos	249
Paralelismo intrapartición con consultas que hacen referencia a apodos - ejemplos de planes de acceso	250
Paralelismo interparticiones con consultas que hacen referencia a apodos	251
Habilitación del paralelismo interparticiones con consultas que hacen referencia a apodos	254
Paralelismo interparticiones con consultas que hacen referencia a apodos - ejemplos de planes de acceso	254
Grupos de particiones computacionales	257
Definición de un grupo de particiones computacionales	257
Paralelismo interparticiones con consultas que hacen referencia a apodos - expectativas de rendimiento	258
Paralelismo mixto con consultas que hacen referencia a apodos	258
Habilitación del paralelismo mixto con consultas que hacen referencia a apodos	259
Paralelismo mixto con consultas que hacen referencia a apodos - ejemplos de planes de acceso	259

Capítulo 25. Proceso asíncrono de consultas federadas 261

Proceso asíncrono de consultas federadas - ejemplos	261
Optimización de asincronía	262
Planes de acceso sin asincronía	262
Planes de acceso optimizados para asincronía	262
Planes de acceso - Ejemplos	263
Control del consumo de recursos	267
Habilitación de la optimización de asincronía	267
Consideraciones sobre los ajustes para la optimización de asincronía	268
Restricciones sobre la optimización de asincronía	268
Determinación de si la optimización de asincronía se aplica a una consulta	269

Capítulo 26. Optimización global 271

Características de servidor que afectan a la optimización global	271
Proporción relativa de velocidad de CPU	272
Proporción relativa de velocidad de E/S	272
Velocidad de comunicaciones entre el servidor federado y el origen de datos	273
Secuencia de clasificación de origen de datos	273
Sugerencias de plan remoto	273
Características de apodo que afectan a la optimización global	274

Especificaciones de índice	274
Estadísticas de catálogo global	275
Actualización de los cambios de fila	276
Actualización de las estadísticas cuando cambian las columnas	276
Análisis de la optimización global	277
Interpretación de decisiones de optimización de planes de acceso	278
Por qué no se evalúa remotamente una unión entre dos apodos del mismo origen de datos	278
Por qué no se evalúa el operador GROUP BY de forma remota	278
Por qué no se evalúa completamente la sentencia de forma remota	278
Por qué un plan generado por el optimizador y completamente evaluado de forma remota tiene un rendimiento mucho peor que la consulta original ejecutada directamente en el origen de datos remoto	279

Capítulo 27. Elementos del supervisor del sistema que afectan al rendimiento 281

Capítulo 28. Tablas de consultas materializadas 283

Tablas de consultas materializadas y sistemas federados - Visión general	283
Creación de una tabla de consultas materializadas federada	284
Restricciones específicas de orígenes de datos para tablas de consultas materializadas	284
Restricciones sobre la utilización de tablas de consultas materializadas con apodos	285

Capítulo 29. Tablas de memoria caché 287

Creación de tablas de memoria caché	288
Modificación de valores para tablas de consulta materializadas	289
Adición de tablas de consulta materializadas a una tabla de memoria caché	290
Direccionamiento de consultas a tablas de memoria caché	291
Habilitación e inhabilitación de los valores de memoria caché de réplica	291
Eliminación de tablas de consulta materializadas de una tabla de memoria caché	292
Eliminación de tablas de memoria caché	293

Capítulo 30. Seguridad para servidores federados 295

Capítulo 31. Contextos fiables federados y conexiones fiables 297

Ventajas de las conexiones fiables federadas	298
Tipos de conexiones fiables federadas	299
API para conexiones fiables federadas	300
Caso de ejemplo para la implementación de conexiones fiables federadas	301

Caso de ejemplo: conexiones fiables federadas de extremo a extremo, sin correlaciones de usuario	301
Código de muestra para casos de ejemplo de conexión fiable federada de extremo a extremo	304
Caso de ejemplo: conexiones fiables federadas de extremo a extremo, con correlaciones de usuario.	304
Caso de ejemplo: conexiones fiables federadas de salida	307
Correlaciones de usuario y conexiones fiables federadas.	311

Capítulo 32. Control de acceso basado en etiquetas (LBAC) y sistemas federados. 315

Capítulo 33. Depósitos de correlaciones de usuarios externos. . 317

Plugin de correlación de usuario (lenguaje de programación C)	317
Plataformas soportadas para el plugin de correlación de usuario (lenguaje de programación C)	319
Restricciones en el desarrollo de un plugin de correlación de usuario (lenguaje de programación C)	319
Archivo de cabecera (fsumplugin.h) para el plugin de correlación de usuario (lenguaje de programación C)	320
Desarrollo del plugin de correlación de usuario (lenguaje de programación C)	324
Plugin de correlación de usuario de muestra (lenguaje de programación C)	329
Plugin de correlación de usuario (lenguaje de programación Java)	330
Clases para el plugin de correlación de usuario (lenguaje de programación Java)	332
Plugin de correlación de usuario de muestra (lenguaje de programación Java)	336
Desarrollo del plugin de correlación de usuario (lenguaje de programación Java)	337

Capítulo 34. Seguridad de Oracle en un sistema federado 345

Seguridad de etiqueta de Oracle	345
Autenticación proxy de Oracle y contextos fiables federados.	345

Capítulo 35. Soporte de origen de datos para opciones federadas. . . . 347

Capítulo 36. Guía de consulta de opciones para orígenes de datos . . . 349

Información de consulta sobre opciones de BioRS	349
Información de consulta sobre opciones de bases de datos de DB2	353
Información de consulta sobre opciones de Excel	360

Información de consulta sobre opciones de Informix	361
Información de consulta sobre opciones de JDBC	367
Información de consulta sobre opciones de Microsoft SQL Server.	374
Información de consulta sobre opciones de ODBC	379
Información de consulta sobre opciones de Oracle	385
Información de consulta sobre opciones de Script	390
Información de consulta sobre opciones de Sybase	396
Información de consulta sobre opciones de Teradata	401
Información de consulta sobre opciones de archivos con estructura de tabla	405
Información de consulta sobre opciones de servicios web	408
Información de consulta sobre opciones de XML	415

Capítulo 37. Vistas de la tabla de catálogo global que contienen información federada 423

Capítulo 38. Opciones de correlaciones de funciones para sistemas federados. 427

Capítulo 39. Tipos de servidor válidos en las sentencias de SQL 429

Capítulo 40. Correlaciones de tipos de datos 431

Correlaciones de tipos de datos directas por omisión	431
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows.	431
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i	432
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE	433
Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS .	433
Correlaciones de tipos de datos directas por omisión para orígenes de datos Informix	434
Correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC	435
Correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server.	437
Correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC	438
Correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8.	439
Correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase.	440
Correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata	441

Correlaciones de tipos de datos directas de ejemplo	442	Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server.	454
Correlaciones de tipos de datos inversas por omisión	444	Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server.	455
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows	445	Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8	455
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i	446	Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8	455
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE	447	Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC	456
Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS	447	Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC	456
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix	448	Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase	457
Correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC	449	Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase	457
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server.	450	Tipos de datos admitidos para orígenes de datos no relacionales	458
Correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC	450	Documentación del producto	461
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8.	451	Cómo leer los diagramas de sintaxis	463
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase	452	Documentación accesible	465
Correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata	453	Avisos	467
Correlaciones de tipos de datos por omisión Unicode	453	Marcas registradas.	469
Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC	453	Índice.	471
Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC	454		

Capítulo 1. Sistemas federados

Un *sistema federado* es un tipo especial de sistema de gestión de bases de datos (DBMS) distribuidas. Un sistema federado consta de una instancia de DB2 que actúa como servidor federado, una base de datos que actúa como base de datos federada, uno o varios orígenes de datos y clientes (usuarios y aplicaciones) que acceden a la base y a los orígenes de datos.

En un sistema federado puede enviar solicitudes distribuidas a varios orígenes de datos dentro de una única sentencia SQL. Por ejemplo, puede unir datos ubicados en una tabla DB2, una tabla Oracle y un archivo codificado en XML en una única sentencia SQL. La imagen siguiente muestra los componentes de un sistema federado y un ejemplo de los orígenes de datos a los que puede acceder.

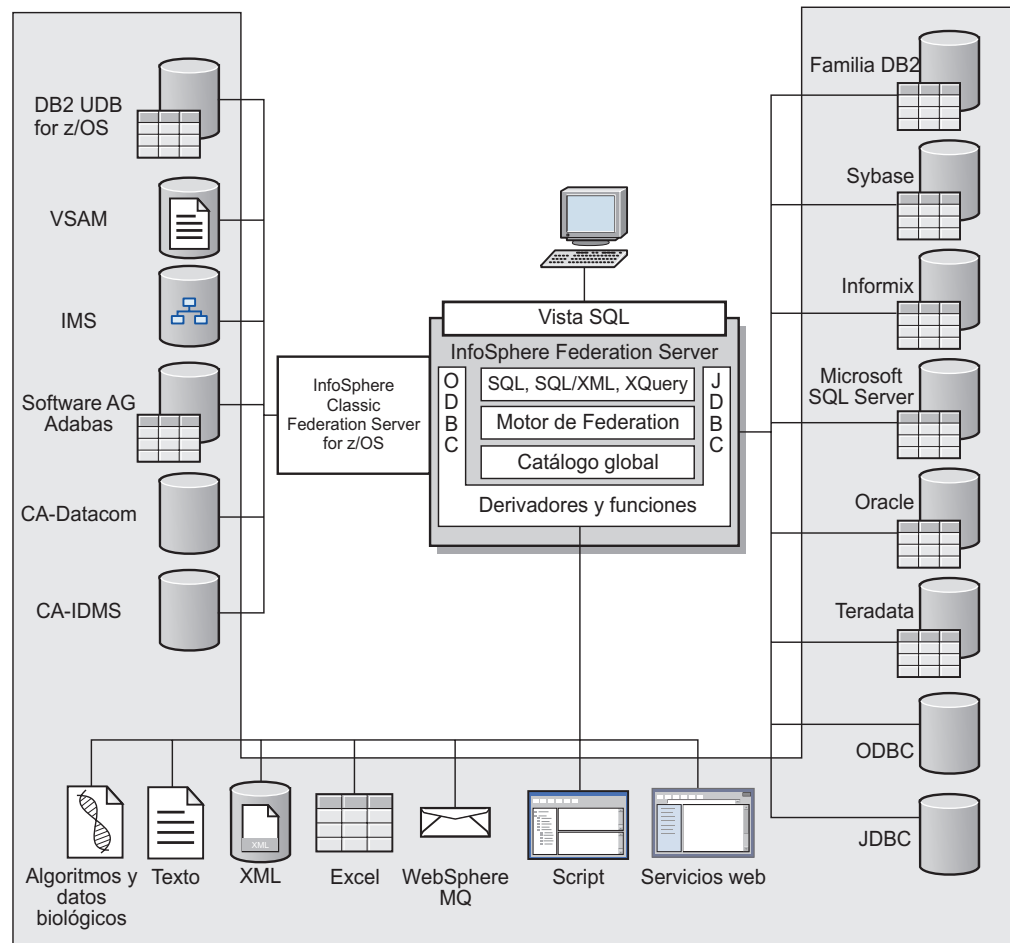


Figura 1. Los componentes de un sistema federado

La potencia de un sistema federado se basa en su capacidad para:

- Correlacionar datos de tablas locales y orígenes de datos remotos, como si todos los datos se almacenaran localmente en la base de datos federada
- Actualizar datos en orígenes de datos relacionales, como si todos los datos se almacenaran en bases de datos federadas

- Mover datos desde los orígenes de datos relacionales y hacia estos orígenes de datos
- Aprovechar la potencia de proceso de orígenes de datos, enviando solicitudes a los orígenes de datos para que se procesen
- Compensar limitaciones SQL en los orígenes de datos procesando partes de una solicitud distribuida en el servidor federado

Orígenes de datos soportados

Son muchos los orígenes de datos a los que puede acceder utilizando un sistema federado.

IBM® InfoSphere Federation Server da soporte a los orígenes de datos que aparecen en las tablas siguientes. La primera tabla lista los requisitos para el software de cliente de datos. El software de cliente debe adquirirse por separado a menos que se especifique lo contrario.

Debe instalar el software de cliente para los orígenes de datos a los que desea acceder. El software de cliente debe instalarse en el mismo sistema que IBM InfoSphere Federation Server. También necesita el Java™ SDK adecuado para utilizar algunas herramientas como, por ejemplo, el Centro de control de DB2 y para crear y ejecutar aplicaciones Java, incluyendo procedimientos almacenados y funciones definidas por el usuario.

Para obtener la información más reciente, consulte la página Data source requirements en la Web.

Tabla 1. Orígenes de datos soportados, requisitos de software de cliente y soporte en sistemas operativos de 32 bits.

			Arquitectura de hardware y sistema operativo de 32 bits	
			X86-32	X86-32
Origen de datos	Versiones que reciben soporte	Software de cliente	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
BioRS	5.2, 5.3	Ninguno	S	S
DB2 para Linux, UNIX® y Windows	8.1.x, 8.2.x, 9.1, 9.5, 9.7	Ninguno	S	S
DB2 para z/OS	7.x, 8.x, 9.x	DB2 Connect V9.7	S	S
DB2 para System i	5.2, 5.3, 5.4, 6.1	DB2 Connect V9.7	S	S
DB2 Server para VSE y VM	7.2 , 7.4	DB2 Connect V9.7	S	S
Archivos planos		Ninguno	S	S

Tabla 1. Orígenes de datos soportados, requisitos de software de cliente y soporte en sistemas operativos de 32 bits. (continuación)

			Arquitectura de hardware y sistema operativo de 32 bits	
			X86-32	X86-32
Origen de datos	Versiones que reciben soporte	Software de cliente	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
Informix	Informix XPS 8.50, 8.51 e Informix IDS IDS 7.31, IDS 9.40, IDS 10.0, 11.5, 11.10	Informix Client SDK versión 2.81.TC2 o posterior; se necesita la versión 3.0 para SLES 10 on Power	S	S
JDBC	3.0 o posterior	Controladores JDBC compatibles con JDBC 3.0 o posterior	S	S
Microsoft® Excel	2000, 2002, 2003, 2007	Ninguno		S
Microsoft SQL Server	Microsoft SQL Server 2000/SP4, 2005, 2008	Para Windows, el controlador ODBC 3.0 (o posterior) de Microsoft SQL Server Client. Para Unix, DataDirect ODBC 5.3	S	S
MQ	MQ7	MQ7	S	S
ODBC	3.0	Controladores ODBC compatibles con ODBC 3.0, o posterior**	S	S
OLE DB	2.7, 2.8	OLE DB 2.0 o posterior	S	S
Oracle	10g, 10gR2, 11g, 11gR1	Ciente de Oracle NET 10.0 - 10.1, 10.2.0.1 con el parche 3807408, 10.1.0.3 con el parche 3807408, 11.1.0.6.0	S	S
Sybase	Sybase ASE 12.5, 15.0	Sybase Open Client 12.5 - 15.0	S	S

Tabla 1. Orígenes de datos soportados, requisitos de software de cliente y soporte en sistemas operativos de 32 bits. (continuación)

			Arquitectura de hardware y sistema operativo de 32 bits	
			X86-32	X86-32
Origen de datos	Versiones que reciben soporte	Software de cliente	Linux®, RedHat Enterprise Linux (RHEL), SUSE	Windows®
Teradata	2.5, 2.6, 12	Shared Common Components for Internationalization for Teradata (tdicu) versión 1.01 o posterior, Teradata Generic Security Services (TeraGSS) versión 6.01 o posterior y el software de los sistemas operativos siguientes: Para Windows, el cliente Teradata TTU 8.0 o posterior y la biblioteca API de Teradata CLIV2 4.8.0 o posterior Para UNIX y Linux Teradata Call-Level Interface Versión 2 CLIV2 Release 4.8.0 o posterior	S	S
Servicios Web	WSDL 1.0, 1.1 SOAP 1.0, 1.1	Ninguno	S	S
XML	XML1.0, XML1.1	Ninguno	S	S

** Se puede utilizar ODBC para acceder a RedBrick 6.20cu5 y InfoSphere Classic Federation V8.2 y anteriores, entre otros orígenes de datos.

Tabla 2. Soporte en sistemas operativos de 64 bits.

Arquitectura de hardware de 64 bits	X86-64	X86-64	Power	Itanium®	Power	Sparc	zSeries
Sistema operativo	Linux RHEL SUSE	Windows	AIX	HP-UX	Linux RHEL SUSE	Solaris	Linux RHEL SUSE
Origen de datos							
BioRS	S	S	S	S	S	S	S

Tabla 2. Soporte en sistemas operativos de 64 bits. (continuación)

Arquitectura de hardware de 64 bits	X86-64	X86-64	Power	Itanium®	Power	Sparc	zSeries
Sistema operativo	Linux RHEL SUSE	Windows	AIX	HP-UX	Linux RHEL SUSE	Solaris	Linux RHEL SUSE
Origen de datos							
DB2 para Linux, UNIX y Windows	S	S	S	S	S	S	S
DB2 para z/OS	S	S	S	S	S	S	S
DB2 para System i	S	S	S	S	S	S	S
DB2 Server para VSE y VM	S	S	S	S	S	S	S
Informix	S		S	S	S	S	S
JDBC	S	S	S	S	S	S	S
Microsoft Excel							
Microsoft SQL Server	S	S	S	S		S	
MQ	S	N	S	S	S	S	S
ODBC	S	S	S***	S		S***	S
OLE DB		S		S			
Oracle	S	S	S	S	S	S	S
Script	S	S	S	S	S	S	S
Sybase	S		S	S	S	S	
Teradata	S		S	S		S	
Servicios Web	S	S	S	S	S	S	S
XML	S	S	S	S	S	S	S

*** Se puede utilizar ODBC para acceder a RedBrick 6.20cu5 y IBM InfoSphere Classic Federation Server for z/OS utilizando clientes de 32 bits y 64 de bits.

El servidor federado

En un sistema federado se hace referencia al servidor DB2 como el servidor federado. Puede configurarse el número de instancias de DB2 que se desee para que funcionen como servidores federados. Puede utilizar instancias de DB2 existentes como servidores federados o puede crear nuevas instancias específicamente para el sistema federado.

La instancia de DB2 que gestiona el sistema federado se denomina servidor dado que responde a las peticiones de los usuarios finales y de las aplicaciones cliente. Con frecuencia, el servidor federado envía partes de las peticiones que recibe a los orígenes de datos para su proceso. Una operación de desplazamiento es una

operación que se procesa de forma remota. La instancia de DB2 que gestiona el sistema federado se denomina servidor federado, aunque actúe como cliente cuando envía las peticiones a los orígenes de datos.

Como cualquier otro servidor de aplicaciones, el servidor federado es una instancia del gestor de bases de datos. Los procesos de aplicación se conectan y envían peticiones a la base de datos que está dentro del servidor federado. Sin embargo, existen dos características principales que lo diferencian de otros servidores de aplicaciones:

- Un servidor federado está configurado para recibir peticiones que podrían destinarse total o parcialmente a los orígenes de datos. El servidor federado distribuye esas peticiones a los orígenes de datos.
- Como otros servidores de aplicaciones, un servidor federado utiliza protocolos de comunicación DRDA (mediante TCP/IP) para comunicarse con instancias de la familia DB2. Sin embargo, a diferencia de otros servidores de aplicaciones, un servidor federado utiliza el cliente nativo del origen de datos para acceder al origen de datos. Por ejemplo, un servidor federado utiliza Sybase Open Client para acceder a orígenes de datos Sybase y un controlador ODBC de Microsoft SQL Server para acceder a orígenes de datos Microsoft SQL Server.

Qué es un origen de datos

En un sistema federado, un *origen de datos* puede ser una base de datos relacional (como por ejemplo Oracle o Sybase) o un origen de datos no relacional (como por ejemplo un archivo codificado en XML).

Por medio de algunos orígenes de datos, puede acceder a otros orígenes de datos. Por ejemplo, con el derivador de ODBC puede acceder a orígenes de datos IBM InfoSphere Classic Federation Server for z/OS tales como DB2 para z/OS, IMS, CA-IDMS, CA-Datcom, Software AG Adabas y VSAM.

El método, o protocolo, que se utiliza para acceder a un origen de datos depende del tipo de éste. Por ejemplo, DRDA se utiliza para acceder a orígenes de datos DB2 para z/OS.

Los orígenes de datos son autónomos. Por ejemplo, el servidor federado puede enviar consultas a orígenes de datos Oracle al mismo tiempo que aplicaciones Oracle acceden a estos orígenes de datos. Un sistema federado no monopoliza ni restringe el acceso al resto de orígenes de datos, más allá de restricciones de integridad y bloqueo.

La base de datos federada

Para usuarios finales y aplicaciones cliente, los orígenes de datos aparecen como una única base de datos colectiva en el sistema DB2. Los usuarios y las aplicaciones interactúan con la *base de datos federada* gestionada por el servidor federado.

La base de datos federada contiene un catálogo del sistema que almacena información sobre los datos. El catálogo del sistema de la base de datos federada contiene entradas que identifican los orígenes de datos y sus características. El servidor federado consulta la información que está almacenada en el catálogo del sistema de la base de datos federada y el derivador del origen de datos para determinar cuál es el mejor plan para procesar las sentencias de SQL.

El sistema federado procesa sentencias de SQL como si los datos de los orígenes de datos fuesen tablas relacionales ordinarias o vistas dentro de la base de datos federada. Como resultado de ello:

- El sistema federado puede correlacionar datos relacionales con datos en formatos no relacionales. Esto también se aplica cuando los orígenes de datos utilizan distintos dialectos de SQL o cuando no dan soporte a SQL.
- Las características de la base de datos federada tienen prioridad cuando existen diferencias entre las características de la base de datos federada y las características de los orígenes de datos. Los resultados de la consulta se ajustan a la semántica de DB2, incluso si se utilizan datos de otros orígenes de datos no DB2 para calcular el resultado de la consulta.

Ejemplos:

- La página de códigos que el servidor federado utiliza es diferente a la página de códigos utilizada por el origen de datos. En este caso, los datos de carácter del origen de datos se convierten en base a la página de códigos utilizada por la base de datos federada, cuando los datos se devuelven a un usuario federado.
- La secuencia de clasificación que el servidor federado utiliza es diferente a la secuencia de clasificación utilizada por el origen de datos. En este caso, cualquier operación de clasificación en datos de carácter se realiza en el servidor federado en lugar de en el origen de datos.

Derivadores y módulos de derivador

Los *derivadores* son mecanismos mediante los cuales la base de datos federada interactúa con orígenes de datos. La base de datos federada utiliza rutinas almacenadas en una biblioteca llamada *módulo de derivador* para implementar un derivador.

Estas rutinas permiten a la base de datos federada realizar operaciones tales como conectar con un origen de datos y recuperar datos de la misma interactivamente. Normalmente, el propietario de la instancia federada utiliza la sentencia CREATE WRAPPER para registrar un derivador en la base de datos federada. Puede registrar un derivador como protegido o fiable utilizando la opción de derivador DB2_FENCED.

Debe crear un derivador para cada tipo de origen de datos al que desee acceder. Por ejemplo, desea acceder a tres tablas de base de datos DB2 para z/OS, una tabla de DB2 para System i, dos tablas de Informix y una vista de Informix. En este caso, necesita crear un derivador para los objetos de origen de datos DB2 y un derivador para los objetos de origen de datos Informix. Después de registrar estos derivadores en la base de datos federada, puede utilizar estos derivadores para acceder a otros objetos desde esos orígenes de datos. Por ejemplo, puede utilizar el derivador DRDA con todos los objetos de origen de datos de la familia DB2-DB2 Database para Linux, UNIX y Windows, DB2 para z/OS, DB2 para System i y DB2 Server para VM y VSE.

Para identificar los datos específicos (nombre, ubicación, etc.) de cada objeto de origen de datos, debe utilizar las definiciones y apodos del servidor.

Un derivador realiza muchas tareas. A continuación se indican algunas de ellas:

- Se conecta con el origen de datos. El derivador utiliza la API de conexión estándar del origen de datos.
- Envía consultas al origen de datos.

- Para los orígenes de datos que dan soporte a SQL, la consulta se envía en SQL.
- Para los orígenes de datos que no dan soporte a SQL, la consulta se traduce al lenguaje de consulta nativo del origen de datos o se convierte en una serie de llamadas de API del origen de datos.
- Recibe los conjuntos de resultados del origen de datos. El derivador utiliza las API estándar del origen de datos para recibir los conjuntos de resultados.
- Responde a consultas de base de datos federada sobre las correlaciones de tipo de datos por omisión para un origen de datos. El derivador contiene las correlaciones de tipos por omisión que se utilizan cuando se crean apodos para un objeto de origen de datos. Para los derivadores relacionales, las correlaciones de tipos de datos que el usuario crea alteran temporalmente las correlaciones de tipos de datos por omisión. Las correlaciones de tipos de datos definidos por el usuario se almacenan en el catálogo global.
- Responde a consultas de base de datos federada sobre las correlaciones de funciones por omisión para un origen de datos. La base de datos federada necesita información de correlación de tipo de datos con la finalidad de planificar las consultas. El derivador contiene información que la base de datos federada necesita para determinar si las funciones de DB2 se correlacionan con funciones del origen de datos y cómo se correlacionan las funciones. El Compilador de SQL utiliza esta información para determinar si el origen de datos es capaz de realizar las operaciones de consulta. Para los derivadores relacionales, las correlaciones de funciones que crea el usuario alteran temporalmente las correlaciones de tipos de funciones por omisión. Las correlaciones de funciones definidas por el usuario se almacenan en el catálogo global.

Las *opciones de derivador* se utilizan para configurar el derivador o para definir el modo en que IBM InfoSphere Federation Server utiliza el derivador.

Cómo interactuar con un sistema federado

Puesto que la base de datos federada es una base de datos de la familia DB2, puede interactuar con ella de varias formas.

Puede interactuar con un sistema federado utilizando cualquiera de estos métodos:

- El procesador de línea de mandatos (CLP) de DB2
- La GUI del Centro de control de DB2
- Programas de aplicación
- Herramientas de la familia DB2
- Proveedores de servicios Web

Los pasos descritos en la documentación federada proporcionan los mandatos y sentencias de SQL que se pueden entrar en el procesador de línea de mandatos de DB2. En la documentación se indica cuándo las tareas pueden realizarse por medio de la GUI del Centro de control de DB2. Puesto que la GUI del Centro de control de DB2 es intuitiva, en esta documentación no se incluyen los pasos para realizar esas tareas por medio del Centro de control de DB2.

Centro de control de DB2

La GUI del Centro de control de DB2 le permite realizar la mayoría de las tareas necesarias para instalar, configurar y modificar el sistema federado. El Centro de control de DB2 utiliza paneles-recuadros de diálogo y asistentes-para guiarle a lo largo de una tarea.

Los paneles del Centro de control de DB2 contienen ayuda interactiva cuando el ratón se sitúa sobre un control como, por ejemplo, un recuadro de lista o un botón de mandato. Además, cada panel dispone de un botón de ayuda que proporciona información acerca de la tarea del panel y también dispone de enlaces con conceptos relacionados e información de consulta.

Puede utilizar un asistente para crear los objetos federados o puede crear cada objeto individualmente.

Utilice el Centro de control de DB2 para configurar el acceso a servicios Web y a orígenes de datos XML. Las características incorporadas al Centro de control de DB2 simplifican los pasos necesarios para configurar el servidor federado para acceder a estos orígenes de datos.

La GUI del Centro de control de DB2 es la forma más sencilla de realizar las tareas esenciales de configuración de orígenes de datos:

- Crear los derivadores y definir las opciones de derivador
- Especificar las variables de entorno del origen de datos
- Crear las definiciones de servidor y definir las opciones de servidor
- Crear las correlaciones de usuarios y definir las opciones de usuario
- Crear los apodos y definir las opciones de apodo u opciones de columna

Después de configurar el servidor federado para acceder a los orígenes de datos, puede utilizar el Centro de control de DB2 para:

- Modificar la configuración del origen de datos
- Supervisar el estado de los apodos y servidores
- Mantener estadísticas actuales para los apodos
- Crear y modificar tablas de antememoria
- Especificar restricciones informativas sobre apodos
- Crear tablas remotas mediante IBM InfoSphere Federation Server utilizando DDL transparente

Programas de aplicación

Las aplicaciones no necesitan ninguna codificación especial para poder utilizarlas con los datos federados. Las aplicaciones acceden al sistema exactamente igual que cualquier otra aplicación cliente de DB2. Las aplicaciones intercambian información con la base de datos que está dentro del servidor federado.

Para obtener datos de los orígenes de datos, las aplicaciones envían consultas en SQL de DB2 a la base de datos federada. La base de datos federada distribuye entonces las consultas a los orígenes de datos apropiados, recopila los datos solicitados y devuelve estos datos a las aplicaciones. Sin embargo, puesto que la base de datos federada interactúa con los orígenes de datos mediante apodos, debe tener presente lo siguiente:

- Las restricciones de SQL que se aplican cuando se trabaja con apodos.
- Cómo ejecutar operaciones sobre objetos con apodos.

Herramientas de la familia DB2

Puede interactuar con una base de datos federada utilizando herramientas de sistema principal y de rango medio.

Las herramientas de sistema principal y de rango medio que puede utilizar para interactuar con una base de datos federada incluyen:

- Herramientas de DB2 que mejoran el rendimiento y la gestión para DB2 para z/OS y DB2 para Linux, UNIX y Windows
- SQL interactivo (STRSQL) en DB2 para System i

IBM Optim Data Studio

IBM Optim Data Studio es una solución exhaustiva de gestión de datos que puede utilizarse para trabajar con la información a la que se accede con las funciones federadas de IBM InfoSphere Federation Server.

Puede utilizar las herramientas de Optim Data Studio para tareas de configuración, diseño y desarrollo.

- Para configuración: IBM Optim Database Administrator for DB2 Linux, UNIX, and Windows

Optim Database Administrator facilita la colaboración entre administradores de base de datos, arquitectos y desarrolladores y simplifica el proceso de identificación, análisis e implementación de cambios de esquema de base de datos en DB2 para Linux, UNIX y Windows.

- Para diseño: IBM InfoSphere Data Architect

InfoSphere Data Architect es un entorno de desarrollo exhaustivo que permite modelar datos, relacionar e integrar activos de datos y desarrollar aplicaciones de base de datos. Con InfoSphere Data Architect, puede buscar y correlacionar relaciones entre diversos orígenes de datos, crear scripts que representen dichas relaciones y, a continuación, desplegar scripts para servidores federados, no federados, locales o remotos.

- Para desarrollo: IBM Optim Development Studio

Optim Development Studio ofrece un completo entorno de desarrollo y prueba para la creación de objetos de base de datos, consultas, lógica de base de datos y aplicaciones pureQuery.

Para obtener más información acerca de Optim Data Studio, consulte Integrated Data Management Information Center, que se encuentra en la dirección <http://publib.boulder.ibm.com/infocenter/idm/v2r2/index.jsp>.

Proveedores de servicios Web

Puede interactuar con una base de datos federada mediante proveedores de servicios Web.

Dentro de sistemas federados, un derivador de servicios Web está disponible para acceder a servicios Web con sentencias de SQL en apodos y vistas que invocan servicios Web. Puede crear un derivador de servicios Web y apodos que especifiquen entrada a los servicios Web y accedan a la salida desde el servicio Web con sentencias SELECT.

Catálogo del sistema de bases de datos federadas

El catálogo del sistema de bases de datos federadas contiene información acerca de los objetos de la base de datos federada e información acerca de los objetos de los orígenes de datos.

En una base de datos federada, el catálogo se denomina catálogo global, pues contiene información acerca de todo el sistema federado. El optimizador de consultas de DB2 utiliza la información del catálogo global y del derivador de origen de datos para planificar la mejor manera de procesar sentencias de SQL. La información almacenada en el catálogo global incluye información remota y local, como por ejemplo nombres de columna, tipos de datos de columna, valores por omisión de columna, información de índice e información de estadísticas.

La información de catálogo remota es la información o el nombre que utiliza el origen de datos. La información de catálogo local es la información o el nombre que utiliza la base de datos federada. Por ejemplo, imaginemos que una tabla remota incluye una columna cuyo nombre es *EMPNO*. El catálogo global almacenaría el nombre de la columna remota como *EMPNO*. A menos que designe un nombre distinto, el nombre de la columna local se almacenará como *EMPNO*. Puede cambiar el nombre de la columna local por *Empleado_número*. Los usuarios que envíen consultas que incluyan esta columna utilizarán *Empleado_número* en sus consultas en lugar de *EMPNO*. Utilice la sentencia ALTER NICKNAME para cambiar el nombre local de las columnas del origen de datos.

Para orígenes de datos relacionales y no relacionales, la información almacenada en el catálogo global incluye tanto información remota como local.

Para ver la información de tabla de origen de datos que está almacenada en el catálogo global, consulte las vistas de catálogo SYSCAT.TABLES, SYSCAT.NICKNAMES, SYSCAT.TABOPTIONS, SYSCAT.INDEXES, SYSCAT.INDEXOPTIONS, SYSCAT.COLUMNS y SYSCAT.COLOPTIONS en la base de datos federada.

El catálogo global también incluye información acerca de los orígenes de datos. Por ejemplo, el catálogo global incluye información que el servidor federado utiliza para conectarse con el origen de datos y para correlacionar las autorizaciones de los usuarios federados con las autorizaciones de los usuarios del origen de datos. El catálogo global contiene atributos acerca del origen de datos que el usuario establece explícitamente, como por ejemplo, las opciones de servidor.

Compilador de SQL

El compilador de SQL de DB2 recopila información para ayudarle a procesar consultas.

Para obtener datos desde orígenes de datos, los usuarios y las aplicaciones envían consultas en SQL a la base de datos federada. Cuando se envía una consulta, el compilador de SQL de DB2 consulta información del catálogo global y del derivador de orígenes de datos para ayudarle a procesar la consulta. Esto incluye información sobre la conexión al origen de datos, información de servidor, correlaciones, información de índice y estadísticas de proceso.

Optimizador de consultas

Como parte del proceso del compilador de SQL, el optimizador de consultas analiza una consulta. El compilador desarrolla estrategias alternativas, llamadas planes de acceso, para procesar la consulta.

Los planes de acceso pueden llamar a la consulta para que ésta:

- La procesen los orígenes de datos.
- La procese el servidor federado.
- La procesen en parte los orígenes de datos y en parte el servidor federado.

El optimizador de consultas evalúa los planes de acceso principalmente en base a la información sobre las capacidades y los datos de la base de datos. El derivador y el catálogo global contienen esta información. El optimizador de consultas descompone la consulta en segmentos que se llaman fragmentos de consulta. Por lo general, es más eficaz enviar un fragmento de consulta a un origen de datos, si ésta puede procesar el fragmento. Sin embargo, el optimizador de consultas tiene en cuenta otros factores, tales como:

- El volumen de datos que se deben procesar.
- La velocidad de proceso del origen de datos.
- El volumen de datos que devolverá el fragmento.
- El ancho de banda de las comunicaciones.
- El hecho de si en el servidor federado existe o no una tabla de consulta materializada utilizable que represente el mismo resultado de consulta.

El optimizador de consultas genera alternativas de plan de acceso para procesar un fragmento de consulta. Las alternativas de plan realizan cantidades variables de trabajo localmente en el servidor federado y en los orígenes de datos remotos. Puesto que el optimizador de consultas está basado en los costes, asigna costes de consumo de recursos a las alternativas de plan de acceso. A continuación, el optimizador de consultas elige el plan que mejor procesará la consulta con el menor consumo de recursos.

Si cualquiera de los fragmentos se van a procesar mediante orígenes de datos, la base de datos federada envía estos fragmentos a los orígenes de datos. Una vez los orígenes de datos procesan los fragmentos, los resultados se recuperan y se devuelven a la base de datos federada. Si la base de datos federada ha realizado cualquier parte del proceso, éste combina sus resultados con los resultados recuperados desde el origen de datos. La base de datos federada, a continuación, devuelve todos los resultados al cliente.

Compensación

La capacidad de IBM InfoSphere Federation Server para procesar SQL que no está soportado por un origen de datos se denomina *compensación*.

La base de datos federada no envía un fragmento de consulta si el origen de datos no puede procesarla o si el servidor federado puede procesarla más rápidamente de lo que puede hacerlo el origen de datos. Por ejemplo, imaginemos que el dialecto SQL de un origen de datos no da soporte a una agrupación CUBE en la cláusula GROUP BY. Se envía al servidor federado una consulta que contiene una agrupación CUBE y especifica una tabla contenida en ese origen de datos. La base de datos federada no envía la agrupación CUBE al origen de datos, sino que procesa CUBE por sí misma.

La base de datos federada compensa la falta de funcionalidad en el origen de datos de dos maneras:

- Puede solicitar que el origen de datos utilice una o más operaciones que sean equivalentes a la función de DB2 indicada en la consulta. Por ejemplo, un origen de datos no da soporte a la función cotangente (COT(x)), pero da soporte a la función tangente (TAN(x)). La base de datos federada puede solicitar que el origen de datos realice el cálculo (1/TAN(x)), que es equivalente a la función cotangente (COT(x)).
- Puede devolver el archivo de datos al servidor federado y ejecutar la función localmente.

Para los orígenes de datos relacionales, cada tipo de RDBMS soporta un subconjunto del estándar SQL internacional. Además, algunos tipos de RDBMS soportan estructuras sintácticas de SQL que quedan fuera de ese estándar. Un *dialecto SQL*, es la totalidad de SQL al que un tipo de RDBMS da soporte. Si se encuentra una estructura sintáctica de SQL en el dialecto SQL de DB2, pero no en el dialecto del origen de datos relacional, el servidor federado puede implementar esta estructura sintáctica en beneficio del origen de datos.

La base de datos federada puede compensar las diferencias en dialectos SQL. Un ejemplo de esta capacidad es la cláusula de expresión de tabla común. El SQL de DB2 incluye la cláusula de expresión de tabla común. En esta cláusula, puede especificarse un nombre por el que todas las cláusulas FROM de una selección completa pueden hacer referencia a un conjunto de resultados. El servidor federado procesará una expresión de tabla común para un origen de datos, aunque el dialecto SQL utilizado por el origen de datos no incluya la expresión de tabla común.

Con la compensación, la base de datos federada puede dar soporte a todo el dialecto SQL de DB2 para consultas de orígenes de datos. Incluso los orígenes de datos con soporte débil para SQL o sin soporte de SQL se beneficiarán de la compensación. Con un sistema federado, debe utilizar el dialecto SQL de DB2, excepto en una sesión de paso a través.

Sesiones de paso a través

Puede enviar sentencias de SQL directamente a los orígenes de datos utilizando una modalidad especial denominada *paso a través*.

Las sentencias de SQL deben enviarse en el dialecto de SQL que utiliza el origen de datos. Utilice una sesión de paso a través cuando desee realizar una operación que no sea posible con DB2 SQL/API. Por ejemplo, utilice una sesión de paso a través para crear un procedimiento, para crear un índice o para realizar consultas en el dialecto nativo del origen de datos.

Actualmente, los orígenes de datos que dan soporte al paso a través lo hacen utilizando SQL. En el futuro, es posible que los orígenes de datos puedan dar soporte al paso a través utilizando un lenguaje de origen de datos distinto de SQL.

De forma similar, puede utilizar una sesión de paso a través para realizar acciones que no están soportadas por SQL, tales como determinadas tareas administrativas. Sin embargo, no puede utilizar una sesión de paso a través para realizar todas las tareas administrativas. Por ejemplo, puede crear o eliminar tablas en el origen de datos, pero no puede iniciar ni detener la base de datos remota.

En una sesión de paso a través, puede utilizar SQL estático y dinámico.

Para gestionar las sesiones de paso a través, el servidor federado proporciona las siguientes sentencias de SQL:

SET PASSTHRU

Abre una sesión de paso a través. Cuando emite otra sentencia SET PASSTHRU para iniciar una nueva sesión de paso a través, la sesión de paso a través actual finaliza.

SET PASSTHRU RESET

Finaliza la sesión de paso a través actual.

GRANT (Server Privileges)

Otorga a un usuario, grupo, lista de ID de autorización o PUBLIC la autorización para iniciar sesiones de paso a través para un origen de datos específico.

REVOKE (Server Privileges)

Revoca la autorización para iniciar sesiones de paso a través.

Definiciones de servidor y opciones de servidor

Después de haber creado los derivadores para los orígenes de datos, el propietario de la instancia federada define los orígenes de datos para la base de datos federada.

El propietario de la instancia proporciona un nombre para identificar el origen de datos y otra información relacionada con el origen de datos. Esta información incluye:

- El tipo y la versión del origen de datos
- El nombre de base de datos para el origen de datos (solo para RDBMS)
- Metadatos que son específicos del origen de datos

Por ejemplo, un origen de datos de la familia DB2 puede tener varias bases de datos. La definición debe especificar la base de datos a la que puede conectarse el servidor federado. En cambio, un origen de datos Oracle tiene una sola base de datos, y el servidor federado puede conectarse a la base de datos sin conocer su nombre. El nombre de la base de datos no está incluido en la definición de servidor federado de un origen de datos Oracle.

El nombre y la otra información que el propietario de la instancia proporciona al servidor federado se denominan, colectivamente, *definición de servidor*. Los orígenes de datos responden a las peticiones de datos y son servidores por derecho propio.

Las sentencias CREATE SERVER y ALTER SERVER se utilizan para crear y modificar una definición de servidor.

Parte de la información de una definición de servidor se almacena en forma de *opciones de servidor*. Cuando cree definiciones de servidor, es importante que entienda las opciones que puede especificar acerca del servidor.

Las opciones de servidor se pueden definir de forma que se conserven de una conexión a otra con la base de datos o bien se pueden definir para que sean vigentes durante una sola conexión.

Correlaciones de usuario

Una *correlación de usuario* es una asociación entre un ID de autorización en el servidor federado y la información necesaria para conectar con el origen de datos remoto.

Para crear una correlación de usuario, debe utilizar la sentencia CREATE USER MAPPING. En la sentencia, se especifica el ID de autorización local, el nombre local del servidor de origen de datos remoto que se especifica en la definición de servidor y el ID y contraseña remotos.

Por ejemplo, supongamos que ha creado una definición de servidor para un servidor remoto y ha especificado 'argon' como nombre local para el servidor remoto. Para hacer que Mary pueda acceder al servidor remoto, cree esta correlación de usuario:

```
CREATE USER MAPPING FOR Mary
SERVER argon
OPTIONS (REMOTE_AUTHID 'ID_remoto', REMOTE_PASSWORD 'contraseña_remota')
```

Cuando Mary emite una sentencia de SQL para conectarse con el servidor remoto, el servidor federado realiza estos pasos:

1. Recupera la correlación de usuario de Mary
2. Descifra la contraseña remota 'contraseña_remota' asociada con el servidor remoto
3. Llama al derivador para conectar con el servidor remoto
4. Pasa al derivador el ID remoto 'ID_remoto' y la contraseña remota descifrada
5. Crea una conexión con el servidor remoto para Mary

Por omisión, el servidor federado almacena la correlación de usuario en la vista SYSCAT.USEROPTIONS del catálogo global y cifra las contraseñas remotas. Como alternativa, puede utilizar un depósito externo, por ejemplo, un archivo o un servidor LDAP, para almacenar correlaciones de usuarios. Para proporcionar la interfaz entre el servidor federado y el depósito externo, debe crear un conector de correlación de usuario.

No importa cómo almacene las correlaciones de usuarios, limite con cuidado el acceso a ellas. Si las correlaciones de usuario están comprometidas, los datos de las bases de datos remotas pueden ser vulnerables a actividades no autorizadas.

Apodos y objetos de origen de datos

Un *apodo* es un identificador que se utiliza para identificar el objeto de origen de datos al que desea acceder. Se hace referencia al objeto que identifica un apodo como *objeto de origen de datos*.

Un apodo no es un nombre alternativo para un objeto de origen de datos de la misma forma que un alias es un nombre alternativo. Un apodo es el puntero mediante el que el servidor federado hace referencia al objeto. Normalmente los apodos se definen mediante la sentencia CREATE NICKNAME junto con determinadas opciones de columna de apodo y opciones de apodo.

Cuando una aplicación cliente o un usuario envía una petición distribuida al servidor federado, la petición no necesita especificar los orígenes de datos. En lugar de ello, la petición especifica los objetos de origen de datos utilizando sus apodos. Los apodos están correlacionados con objetos específicos contenidos en el

origen de datos. Estas correlaciones eliminan la necesidad de calificar los apodos con los nombres de los orígenes de datos. La ubicación de los objetos de origen de datos es transparente a la aplicación cliente o al usuario.

Suponga que define el apodo *DEPT* para representar una tabla de bases de datos Informix llamada *NFX1.PERSON*. El servidor federado admite la sentencia `SELECT * FROM DEPT`. Sin embargo, la sentencia `SELECT * FROM NFX1.PERSON` no está permitida desde el servidor federado (excepto en una sesión de paso a través) a menos que haya una tabla local en el servidor federado llamada *NFX1.PERSON*.

Cuando crea un apodo para un objeto de origen de datos, se añaden metadatos acerca del objeto al catálogo global. El optimizador de consultas utiliza estos metadatos, así como la información del derivador, para facilitar el acceso al objeto de origen de datos. Por ejemplo, si un apodo es para una tabla que tiene un índice, el catálogo global contiene información acerca del índice y el derivador contiene las correlaciones entre los tipos de datos de DB2 y los tipos de datos de origen de datos.

Los apodos para objetos que utilizan control de acceso basado en etiquetas (LBAC) no se colocan en la antememoria. Por lo tanto, los datos en el objeto permanecen seguros. Por ejemplo, si utiliza el derivador Oracle (Net8) para crear un apodo en una tabla que utiliza Oracle Label Security (seguridad de etiqueta de Oracle), la tabla se identifica automáticamente como segura. Los datos de apodo resultantes no se pueden poner en la antememoria. Como consecuencia, las tablas de consultas materializadas no se pueden crear en los mismos. La utilización de LBAC garantiza que sólo vean la información los usuarios con los privilegios de seguridad adecuados. Para apodos creados antes de que LBAC estuviera soportado, debe utilizar la sentencia `ALTER NICKNAME` para inhabilitar la colocación en la antememoria. LBAC está soportado tanto por el derivador de DRDA (para orígenes de datos que utilizan DB2 for Linux, UNIX, and Windows versión 9.1 y posteriores) como por el derivador de Net8.

Objetos de origen de datos válidos

Los apodos identifican objetos que están en el origen de datos al que desea acceder. En la tabla siguiente se indican los tipos de objetos para los que puede crear un apodo en un sistema federado.

Tabla 3. Objetos de origen de datos válidos

Origen de datos	Objetos válidos
BioRS	Bancos de datos BioRS
DB2 Database para Linux, UNIX y Windows	Apodos, tablas de consultas materializadas, tablas, vistas
DB2 para System i	Tablas, vistas, archivos P/L (físicos/lógicos) y tipos de tablas
DB2 para VM y VSE	Tablas, vistas
DB2 para z/OS	Tablas, vistas
Informix	Tablas, vistas, sinónimos
JDBC	Todos los tipos de tablas
Microsoft Excel	Archivos .xls (sólo se accede a la primera hoja del libro de trabajo)
Microsoft SQL Server	Tablas, vistas

Tabla 3. Objetos de origen de datos válidos (continuación)

Origen de datos	Objetos válidos
ODBC	Tablas, vistas
Oracle	Tablas, vistas, sinónimos
Script	Scripts
Sybase	Tablas, vistas
Teradata	Tablas, vistas
Archivos con estructura de tabla	Archivos de texto que se ajustan a un determinado formato.
Servicios Web	Operaciones contenidas en un archivo de lenguaje de descripción de servicios Web
Archivos etiquetados XML	Conjuntos de elementos de un documento XML

Opciones de columna de apodo

Puede proporcionar al catálogo global información de metadatos adicional acerca del objeto con apodo. Estos metadatos describen valores en determinadas columnas del objeto de origen de datos. El usuario asigna estos metadatos a parámetros llamados *opciones de columna de apodo*.

Las opciones de columna de apodo indican al derivador que debe manejar los datos de una columna de forma distinta a como normalmente los manejaría. El compilador de SQL y el optimizador de consultas utilizan los metadatos para desarrollar mejores planes para acceder a los datos.

Las opciones de columna de apodo también se utilizan para proporcionar otros elementos de información al derivador. Por ejemplo, para los orígenes de datos XML, se utiliza una opción de columna de apodo para indicar al derivador la expresión XPath que debe utilizar cuando el derivador analice la columna fuera del documento XML.

Con la federación, el servidor DB2 trata el objeto de origen de datos al que hace referencia un apodo como si fuese una tabla de DB2 local. Como resultado, el usuario puede definir opciones de columna de apodo para cualquier objeto de origen de datos para el que se cree un apodo. Algunas opciones de columna de apodo están pensadas para tipos determinados de orígenes de datos y sólo pueden aplicarse a esos orígenes de datos.

Suponga que un origen de datos tiene una secuencia de clasificación que difiere de la secuencia de clasificación de la base de datos federada. Normalmente, el servidor federado no clasificaría ninguna columna que contuviese datos de carácter en el origen de datos. Devolvería los datos a la base de datos federada y realizaría la clasificación localmente. Sin embargo, suponga que los datos de la columna son de tipo carácter (CHAR o VARCHAR) y que sólo contiene caracteres numéricos ('0','1',..., '9'). Puede indicar este hecho asignando el valor 'Y' a la opción de columna de apodo NUMERIC_STRING. Esto proporciona al optimizador de consultas de DB2 la opción de realizar la clasificación en el origen de datos. Si la ordenación se realiza de forma remota, puede evitar la actividad que supone transferir los datos al servidor federado y realizar la clasificación localmente.

Puede definir opciones de columna de apodo para apodos relacionales utilizando las sentencias ALTER NICKNAME. Puede definir opciones de columna de apodo para apodos no relacionales utilizando las sentencias CREATE NICKNAME y ALTER NICKNAME.

Correlaciones de tipos de datos

Los tipos de datos en el origen de datos deben correlacionarse con los tipos de datos DB2 correspondientes de manera que el servidor federado pueda recuperar datos desde los orígenes de datos.

A continuación se muestran algunos ejemplos de correlaciones de tipos de datos por omisión:

- El tipo FLOAT de Oracle se correlaciona con el tipo DOUBLE de DB2
- El tipo DATE de Oracle se correlaciona con el tipo TIMESTAMP de DB2
- El tipo DATE de DB2 para z/OS™ se correlaciona con el tipo DATE de DB2

Para la mayor parte de orígenes de datos, las correlaciones de tipos por omisión se encuentran en los derivadores. Las correlaciones de tipos por omisión para orígenes de datos DB2 están en el derivador DRDA. Las correlaciones de tipos por omisión para Informix están en el derivador INFORMIX, etc.

Para algunas orígenes de datos no relacionales, hay que especificar la información de tipos de datos en la sentencia CREATE NICKNAME. Los tipos de datos DB2 correspondientes deben especificarse para cada columna del objeto de origen de datos cuando se crea el apodo. Cada columna debe correlacionarse con un campo o columna determinados del objeto del origen de datos.

Para los orígenes de datos relacionales, puede alterar temporalmente las correlaciones de tipos de datos definidas por omisión. Por ejemplo, el tipo de datos INTEGER de Informix por omisión está correlacionado con el tipo de datos INTEGER de DB2. Puede alterar temporalmente las correlaciones por omisión y correlacionar el tipo de datos INTEGER de Informix con el tipo de datos DECIMAL(10,0) de DB2.

Correlaciones de funciones

Para que el servidor federado reconozca una función de origen de datos, la función debe correlacionarse con una función equivalente existente en DB2 Database para Linux, UNIX y Windows.

IBM InfoSphere Federation Server proporciona correlaciones por omisión entre funciones de origen de datos existentes y funciones equivalentes de DB2. Para la mayoría de orígenes de datos, las correlaciones de funciones por omisión están en los derivadores. Las correlaciones de funciones por omisión con funciones de DB2 para z/OS se encuentran en el derivador DRDA. Las correlaciones de funciones por omisión con funciones de Sybase se encuentran en el derivador de CTLIB, etc.

Para los orígenes de datos relacionales, puede crear una correlación de funciones cuando desee utilizar una función de origen de datos que el servidor federado no reconoce. La correlación que cree se establecerá entre la función de origen de datos y una función equivalente de DB2 en la base de datos federada. Por lo general, las correlaciones de funciones se utilizan cuando existe una nueva función incorporada o una nueva función definida por el usuario disponible en el origen de datos. Las

correlaciones de funciones también se utilizan cuando no existe una función equivalente de DB2. En este caso, debe crear también una plantilla de función.

Especificaciones de índice

Cuando crea un apodo para una tabla de origen de datos, al catálogo global se añade información acerca de los índices que tiene la tabla de origen de datos. El optimizador de consultas utiliza esta información para acelerar el proceso de las peticiones distribuidas. La información de catálogo acerca de un índice de origen de datos es un conjunto de metadatos y se denomina especificación de índice.

El optimizador de consultas utiliza esta información para acelerar el proceso de las peticiones distribuidas.

Un servidor federado no crea una especificación de índice cuando el usuario crea un apodo para los objetos siguientes:

- Una tabla que no tiene ningún índice
- Una vista, que normalmente no tiene información de índice almacenada en el catálogo remoto
- Un objeto de origen de datos que no tiene un catálogo remoto del que el servidor federado pueda obtener información de índice

Imaginemos que una tabla adquiere un nuevo índice, además de los que ya tenía al crearse el apodo. Puesto que la información de índice se proporciona al catálogo global durante la creación del apodo, el servidor federado no reconoce el nuevo índice. De forma similar, cuando se crea un apodo para una vista, el servidor federado no reconoce la tabla subyacente (ni los índices de ésta) a partir de la cual se ha generado la vista. En estas circunstancias, puede proporcionar la información de índice necesaria al catálogo global. Puede crear una especificación de índice para las tablas que no tienen ningún índice. La especificación de índice indica al optimizador de consultas en qué columna o columnas de la tabla debe buscar para encontrar los datos rápidamente.

Procedimientos almacenados federados

El acceso a procedimientos federados permite a los usuarios de sistemas federados acceder a procedimientos almacenados remotos en orígenes de datos remotos.

Un procedimiento almacenado federado es un procedimiento almacenado local que se correlaciona con un procedimiento almacenado en un origen de datos. Se utiliza la sentencia CREATE PROCEDURE (de origen) para registrar un procedimiento almacenado federado.

Secuencias de clasificación

El orden en el que los datos de carácter se almacenan en una base de datos depende de la estructura de los datos y de la secuencia de clasificación definida para la base de datos.

Suponga que los datos de una base de datos están todos en letras mayúsculas y no contienen caracteres numéricos o especiales. Una clasificación de los datos debería dar como resultado la misma salida, independientemente de si los datos están almacenados en el origen de datos o en la base de datos federada. La secuencia de clasificación utilizada por cada base de datos no debería influir en los resultados de clasificación. De la misma manera, si los datos de la base de datos están todos

en letras minúsculas o son todos caracteres numéricos, una clasificación de los datos debería generar los mismos resultados independientemente de si la clasificación se efectúa realmente.

Si los datos consisten en cualquiera de las siguientes estructuras:

- Una combinación de letras y caracteres numéricos
- Letras tanto mayúsculas como minúsculas
- Caracteres especiales tales como @, #, €

La clasificación de estos datos puede dar como resultado diferentes salidas, si la base de datos federada y el origen de datos utilizan diferentes secuencias de clasificación.

En términos generales, una secuencia de clasificación es un orden definido para datos de carácter que determina si un carácter en particular se clasifica por encima, por debajo o al mismo nivel que otro carácter.

Cómo determinan las secuencias de clasificación los órdenes de clasificación

Una secuencia de clasificación determina el orden de clasificación de los caracteres en un conjunto de caracteres codificado.

Un conjunto de caracteres es el agregado de caracteres que se utilizan en un sistema informático o lenguaje de programación. En un conjunto de caracteres codificado, cada carácter se asigna a un número diferente dentro del rango de 0 a 255 (o el equivalente hexadecimal del mismo). Los números reciben el nombre de elemento de código; las asignaciones de números a caracteres en un conjunto se denominan colectivamente una página de códigos.

Además de asignarse a un carácter, un elemento de código se puede correlacionar con la posición del carácter en un orden de clasificación. En términos técnicos, por tanto, una secuencia de clasificación en la correlación colectiva de los elementos de código de un conjunto de caracteres a las posiciones de orden de clasificación de los caracteres del conjunto. La posición de un carácter se representa mediante un número; este número se denomina el peso del carácter. En la secuencia de clasificación más simple, llamada una secuencia de identidad, los pesos son idénticos a los elementos de código.

Ejemplo: La base de datos ALPHA utiliza la secuencia de clasificación por omisión de la página de códigos EBCDIC. La base de datos BETA utiliza la secuencia de clasificación por omisión de la página de códigos ASCII. Los órdenes de clasificación para series de caracteres en estas dos bases de datos diferirían:

```
SELECT.....  
ORDER BY COL2
```

Clasif. basada en EBCDIC	Clasif. basada en ASCII
COL2	COL2
----	----
V1G	7AB
Y2W	V1G
7AB	Y2W

Ejemplo: De manera similar, las comparaciones de caracteres en una base de datos dependen de la secuencia de clasificación definida para esa base de datos. La base

de datos ALPHA utiliza la secuencia de clasificación por omisión de la página de códigos EBCDIC. La base de datos BETA utiliza la secuencia de clasificación por omisión de la página de códigos ASCII. Las comparaciones de caracteres en estas dos bases de datos generarían resultados distintos:

```
SELECT.....
  WHERE COL2 > 'TT3'
```

Result. basados en EBCDIC	Result. basados en ASCII
COL2	COL2
----	----
TW4	TW4
X82	X82
39G	

Establecimiento de la secuencia de clasificación local para optimizar consultas

Los administradores pueden crear bases de datos federadas con una secuencia de clasificación determinada que coincida con la secuencia de clasificación de un origen de datos.

Para cada definición de servidor de origen de datos, la opción de servidor `COLLATING_SEQUENCE` se establece en 'Y'. Este valor le indica a la base de datos federada que las secuencias de clasificación de la base de datos federada y del origen de datos coinciden.

La secuencia de clasificación de la base de datos federada se establece como parte del mandato `CREATE DATABASE`. Con este mandato, puede especificar una de las siguientes secuencias:

- Una secuencia de identidad
- Una secuencia de sistema (la secuencia utilizada por el sistema operativo que da soporte a la base de datos)
- Una secuencia personalizada (una secuencia predefinida que la base de datos DB2 suministra o que el usuario define)

Supongamos que el origen de datos es DB2 para z/OS. Las clasificaciones que se definen en una cláusula `ORDER BY` las implementa una secuencia de clasificación basada en una página de códigos EBCDIC. Para recuperar datos de DB2 para z/OS clasificados de acuerdo con cláusulas `ORDER BY`, configure la base de datos federada de manera que utilice la secuencia de clasificación predefinida basada en la página de códigos EBCDIC adecuada.

Capítulo 2. Modificación de configuraciones de orígenes de datos

Con el tiempo, es posible que determine que debe modificar las configuraciones de los orígenes de datos para el sistema federado. Por ejemplo, si los orígenes de datos cambian, es posible que deba actualizar las definiciones de servidor, los apodos y las correlaciones de usuario. Es posible que también deba añadir o eliminar acceso a los orígenes de datos desde el sistema federado.

Modificación de derivadores

- Modificación de un derivador (Centro de control de DB2)
- Modificación de un derivador (línea de mandatos de DB2)

Modificación de definiciones y opciones de servidor

- Modificación de definiciones de servidor y de opciones de servidor
- Utilización de opciones de servidor en definiciones de servidor (Centro de control de DB2)
- Utilización de opciones de servidor en definiciones de servidor (línea de mandatos de DB2)

Modificación de correlaciones de usuario

- Modificación de una correlación de usuario (Centro de control de DB2)
- Modificación de una correlación de usuario (línea de mandatos de DB2)

Modificación de apodos

- Modificación de un apodo (Centro de control de DB2)
- Modificación de un apodo (línea de mandatos de DB2)

Eliminación de derivadores, definiciones de servidor, correlaciones de usuario y apodos

- Eliminación de un derivador
- Eliminación de una definición de servidor
- Eliminación de una correlación de usuario
- Eliminación de un apodo

Modificación de un derivador (Centro de control de DB2)

Después de configurar un derivador, puede utilizar la sentencia ALTER WRAPPER para modificar esa configuración basándose en los requisitos del sistema. Puede modificar un derivador desde el Centro de control de DB2 o utilizando la línea de mandatos de DB2.

Antes de empezar

El ID de autorización asociado a la sentencia debe tener autoridad SYSADM o DBADM.

Restricciones

No se puede descartar la opción de derivador DB2_FENCED.

El servidor federado no puede procesar una sentencia ALTER WRAPPER en una unidad de trabajo determinada si dicha unidad de trabajo ya incluye una de las sentencias siguientes:

- Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos que incluya el derivador
- Un cursor abierto en un apodo para una tabla o vista en el origen de datos que incluya el derivador
- Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en el origen de datos que incluya el derivador

Acerca de esta tarea

Esta tarea describe cómo modificar un derivador desde el Centro de control de DB2.

Procedimiento

Para modificar un derivador desde el Centro de control de DB2:

1. Expanda la carpeta Objetos de base de datos federada. Los objetos de derivador se visualizarán en el panel de contenido de la ventana del Centro de control de DB2.
2. Pulse con el botón derecho del ratón en el derivador que desee cambiar y, a continuación, pulse **Modificar** en la lista de acciones. Se abrirá el cuaderno Modificar derivador.
 - a. En la página Valores, efectúe los cambios.
 - b. Pulse en **Establecer variables** para establecer las variables de entorno de origen de datos para el derivador. No se requieren variables de entorno para todos los derivadores.
3. Pulse **Aceptar** para modificar el derivador y cierre el cuaderno Modificar derivador.

Modificación de un derivador - ejemplos

Este tema proporciona ejemplos de las opciones de modificación de derivador mediante la sentencia ALTER WRAPPER.

Para cambiar la opción DB2_FENCED a 'Y' para el derivador denominado drda, emita la sentencia siguiente:

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

Para cambiar la opción MODULE a '/opt/odbc/lib/libodbc.a(odbc.so)' para el derivador denominado odbc, emita la sentencia siguiente:

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

Modificación de un derivador (línea de mandatos de DB2)

Después de configurar un derivador, puede utilizar la sentencia ALTER WRAPPER para modificar esa configuración basándose en los requisitos del sistema. Puede modificar un derivador desde el Centro de control de DB2 o utilizando la línea de mandatos de DB2.

Antes de empezar

El ID de autorización asociado a la sentencia debe tener autoridad SYSADM o DBADM.

Restricciones

No se puede descartar la opción de derivador DB2_FENCED.

El servidor federado no puede procesar una sentencia ALTER WRAPPER en una unidad de trabajo determinada si dicha unidad de trabajo ya incluye una de las sentencias siguientes:

- Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos que incluya el derivador
- Un cursor abierto en un apodo para una tabla o vista en el origen de datos que incluya el derivador
- Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en el origen de datos que incluya el derivador

Acerca de esta tarea

Esta tarea describe cómo modificar un derivador desde la línea de mandatos de DB2. Puede utilizar la sentencia ALTER WRAPPER para añadir, establecer o descartar una o más opciones de derivador.

Procedimiento

Para modificar un derivador desde la línea de mandatos de DB2, emita la sentencia ALTER WRAPPER.

Modificación de definiciones de servidor y opciones de servidor

Debe utilizar la sentencia ALTER SERVER para modificar la definición del servidor. Parte de la información de una definición de servidor se almacena en forma de opciones de servidor. Cuando modifique una definición de servidor, es importante que comprenda las opciones que puede especificar sobre el servidor.

Una definición de servidor identifica un origen de datos para la base de datos federada. La definición de servidor consta de un nombre local y de otra tipo de información sobre ese servidor de orígenes de datos. La definición de servidor es utilizada por el derivador cuando las sentencias de SQL que utilizan apodos se someten a la base de datos federada.

Para los orígenes de datos relacionales, las opciones de servidor pueden establecerse temporalmente mediante la sentencia SET SERVER OPTION. Esta sentencia altera temporalmente el valor de la opción de servidor en la definición de servidor durante una sola conexión con la base de datos federada. Para SQL estático, el uso de la sentencia SET SERVER OPTION sólo afecta a la ejecución de la sentencia SQL estática. El uso de la sentencia SET SERVER OPTION no tiene ningún efecto sobre los planes que genera el optimizador.

Modifique una definición de servidor cuando:

- Actualice a una nueva versión del origen de datos
- Desea realizar el mismo cambio para todas las definiciones de servidor para un tipo de origen de datos específico

- Desea añadir o cambiar una opción de servidor en una definición de servidor existente

Restricciones en la modificación de definiciones de servidor

Cuando se modifique una definición de servidor deben tenerse presentes varias restricciones.

Las siguientes restricciones se aplican a la modificación de definiciones de servidor:

- No se puede especificar un derivador en la sentencia ALTER SERVER que no esté registrado en el servidor federado.
- El servidor federado no puede procesar una sentencia ALTER SERVER en una unidad de trabajo (UOW) determinada bajo ninguna de las siguientes condiciones:
 - La sentencia hace referencia a una sola origen de datos, y la UOW ya incluye una de las sentencias siguientes:
 - Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos
 - Un cursor abierto en un apodo para una tabla o vista en el origen de datos
 - Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en el origen de datos
 - La sentencia hace referencia a una categoría de orígenes de datos (por ejemplo, todas los orígenes de datos de un tipo y versión específicos), y la UOW ya incluye una de las sentencias siguientes:
 - Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en una de los orígenes de datos
 - Un cursor abierto en un apodo para una tabla o vista en una de los orígenes de datos
 - Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en una de los orígenes de datos
- El servidor federado no verifica si la versión de servidor que se ha especificado coincide con la versión de servidor remoto. Especificar una versión de servidor incorrecta puede resultar en errores de SQL al acceder a los apodos que pertenecen a la definición de servidor de DB2. Es más probable que estos errores de SQL se produzcan al especificar una versión de servidor que sea posterior a la versión real del servidor remoto. En ese caso, al acceder a los apodos que pertenecen a la definición de servidor, el servidor federado puede enviar SQL al servidor remoto que este no reconocerá. En la sentencia ALTER SERVER, esta situación se aplica sólo al formato de la sentencia que modifica la versión de servidor (*nombre-servidor* VERSION *versión-servidor*).
- Debe especificar el nombre completo de la opción de servidor. Por ejemplo, no puede especificar la abreviatura DB2_TWO_PHASE. En su lugar, es necesario que especifique el nombre completo de la opción de servidor DB2_TWO_PHASE_COMMIT.

Modificación de la versión de origen de datos en una definición de servidor (Centro de control de DB2)

Se puede modificar una definición de servidor existente para cambiar la versión del origen de datos que utiliza el servidor remoto. Puede modificar una definición de servidor desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

La autorización de ID que emite la sentencia ALTER SERVER debe incluir autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

Consulte el apartado “Restricciones en la modificación de definiciones de servidor” en la página 26.

Procedimiento

Para modificar una definición de servidor desde el Centro de control de DB2:

1. Expanda la carpeta Objetos de base de datos federada. Los objetos de la definición de servidor se visualizarán en el panel de contenido de la ventana del Centro de control de DB2.
2. Pulse con el botón derecho del ratón en la definición de servidor que desee cambiar y, a continuación, pulse **Modificar** en la lista de acciones. Se abrirá el cuaderno Modificar definición de servidor.
3. En la página de Servidor, pulse en la flecha de **Versión** para especificar una versión distinta del origen de datos.
4. Pulse **Aceptar** para modificar la definición de servidor y cierre el cuaderno Modificar definición de servidor.

Modificación de la versión de origen de datos en una definición de servidor (línea de mandatos de DB2)

Se puede modificar una definición de servidor existente para cambiar la versión del origen de datos que utiliza el servidor remoto. Puede modificar una definición de servidor desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

La autorización de ID que emite la sentencia ALTER SERVER debe incluir autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

Consulte el apartado “Restricciones en la modificación de definiciones de servidor” en la página 26.

Procedimiento

Para modificar una definición de servidor desde la línea de mandatos de DB2, emita la sentencia ALTER SERVER.

Ejemplo: Está trabajando con una definición de servidor para un origen de datos de Microsoft SQL Server Versión 6.5. El nombre que ha asignado al servidor en la sentencia CREATE SERVER es `SQLSVR_ASIA`. Si Microsoft SQL Server se actualiza a la Versión 7.0, la sentencia siguiente modificará la definición de servidor:

```
ALTER SERVER SQLSVR_ASIA VERSION 7
```

Modificación de todas las definiciones de servidor para un tipo de origen de datos específico

Puede modificar todas las definiciones de servidor para un tipo de origen de datos determinado con una única sentencia ALTER SERVER. Utilice una única sentencia cuando desee que se aplique el mismo cambio a todas las definiciones de servidor del mismo tipo.

Antes de empezar

La autorización de ID que emite la sentencia ALTER SERVER debe incluir autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

Sólo se pueden establecer o descartar opciones de servidor mediante la sentencia ALTER SERVER para un tipo completo de orígenes de datos si las opciones de servidor fueron añadidas mediante una operación previa de la sentencia ALTER SERVER.

Procedimiento

Para modificar todas las definiciones de servidor para un tipo de origen de datos determinado, emita una única sentencia ALTER SERVER.

Ejemplo: Se han registrado cinco servidores de Sybase en el catálogo global para sus orígenes de datos de Sybase. Siempre que envíe un ID de usuario a alguno de dichos servidores de Sybase para su autenticación, desea que el servidor federado convierta siempre el ID de usuario a mayúsculas. Además, quiere establecer cuánto tiempo deberá esperar el servidor federado una respuesta de estos servidores Sybase a una sentencia de SQL. Debe especificar la cantidad de tiempo en segundos. Puede modificar las cinco definiciones de servidor al mismo tiempo utilizando la sentencia ALTER SERVER siguiente:

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

Utilización de opciones de servidor en definiciones de servidor (Centro de control de DB2)

Existen opciones de servidor generales y opciones de servidor que son específicas para algunos tipos de orígenes de datos. Puede modificar una definición de servidor desde el Centro de control de DB2 o desde el indicador de línea de mandatos para añadir o cambiar una opción de servidor.

Antes de empezar

La autorización de ID que emite la sentencia ALTER SERVER debe incluir autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

Consulte el apartado “Restricciones en la modificación de definiciones de servidor” en la página 26.

Acerca de esta tarea

Las opciones de servidor están establecidas en valores que se conservan durante sucesivas conexiones con el origen de datos. Dichos valores se almacenan en el catálogo del sistema federado.

Procedimiento

Para realizar esta tarea desde el Centro de control de DB2:

1. Expanda la carpeta Objetos de base de datos federada. Los objetos de la definición de servidor se visualizarán en el panel de contenido de la ventana del Centro de control de DB2.
2. Pulse con el botón derecho del ratón en la definición de servidor que desee cambiar y, a continuación, pulse **Modificar** en la lista de acciones. Se abrirá el cuaderno Modificar definición de servidor.
3. En la página de Valores, seleccione la opción de servidor que desee añadir o eliminar.
4. Para las opciones que añada o cambie, especifique el valor de una opción.
5. Pulse **Aceptar** para modificar la definición de servidor y cierre el cuaderno Modificar definición de servidor.

Algunas opciones de servidor son necesarias y no se pueden descartar. Otras opciones de servidor no pueden añadirse si ya se han establecido opciones de servidor específicas.

Modificación temporal de opciones de servidor para orígenes de datos relacionales

La sentencia SET SERVER OPTION altera temporalmente el valor de la opción de servidor en la definición de servidor durante una sola conexión con la base de datos federada. El valor de la alteración temporal no se almacena en el catálogo global.

Acerca de esta tarea

Para SQL estático, el uso de la sentencia SET SERVER OPTION sólo afecta a la ejecución de la sentencia SQL estática. El uso de la sentencia SET SERVER OPTION no tiene ningún efecto sobre los planes que genera el optimizador.

Procedimiento

Para establecer un valor de opción de servidor de forma temporal para un origen de datos relacional, utilice la sentencia SET SERVER OPTION.

Por ejemplo:

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

Jerarquía de los valores de opciones de servidor

Cuando se tiene la misma opción de servidor establecida con un valor para un tipo de origen de datos y establecida con otro valor en un servidor de orígenes de datos específico, existe una jerarquía entre los valores.

Por ejemplo, la opción de servidor PLAN_HINTS se establece en 'Y' para el tipo de origen de datos SYBASE. Sin embargo, la opción de servidor PLAN_HINTS se establece en 'N' en la definición de servidor para un servidor de orígenes de datos Sybase específico PURNELL. El valor del servidor de orígenes de datos específico altera temporalmente el valor del tipo de origen de datos. Esta configuración

provoca que la opción `PLAN_HINTS` se habilite en todos los servidores de orígenes de datos Sybase excepto en PURNELL.

Utilización de opciones de servidor en definiciones de servidor (línea de mandatos de DB2)

Existen opciones de servidor generales y opciones de servidor que son específicas para algunos tipos de orígenes de datos. Puede modificar una definición de servidor desde el Centro de control de DB2 o desde el indicador de línea de mandatos para añadir o cambiar una opción de servidor.

Antes de empezar

La autorización de ID que emite la sentencia `ALTER SERVER` debe incluir autoridad `SYSADM` o `DBADM` en la base de datos federada.

Restricciones

Consulte el apartado “Restricciones en la modificación de definiciones de servidor” en la página 26.

Acerca de esta tarea

Las opciones de servidor están establecidas en valores que se conservan durante sucesivas conexiones con el origen de datos. Dichos valores se almacenan en el catálogo del sistema federado.

Procedimiento

Para efectuar esta tarea desde el indicador de línea de mandatos, emita la sentencia `ALTER SERVER`. Por ejemplo:

- El usuario había creado una definición de servidor para un servidor de Informix utilizando el nombre de servidor `INFMX01`. Ahora le interesa cambiar la opción `DB2_MAXIMAL_PUSHDOWN` a `Y`. La sentencia para modificar la definición de servidor es:

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

- El usuario había creado una definición de servidor para un servidor de Oracle utilizando el nombre de servidor `ORCL99`. Ahora le interesa añadir las opciones `FOLD_ID` y `FOLD_PW` a dicha definición. La sentencia para modificar la definición de servidor es:

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

- Desea establecer el valor del tiempo de espera para cuantos segundos debería esperar el derivador `CTLIB` una respuesta del servidor de Sybase. Utilice la opción de servidor `TIMEOUT` para establecer este valor. La sentencia para modificar la definición de servidor es:

```
ALTER SERVER SYBSERVER OPTIONS (ADD TIMEOUT '60')
```

Modificación de una correlación de usuario (Centro de control de DB2)

Una correlación de usuario es la asociación entre el ID de autorización del servidor federado y el ID de autorización del origen de datos. Las correlaciones de usuario son necesarias para que las peticiones distribuidas puedan enviarse al origen de datos.

Antes de empezar

Si el ID de autorización que emite la sentencia es diferente al ID de autorización que se correlaciona con el origen de datos, entonces el ID de autorización que emite la sentencia debe incluir la autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

El servidor federado no puede procesar una sentencia ALTER USER MAPPING en una unidad de trabajo determinada (UOW) si dicha UOW ya incluye una de las sentencias siguientes:

- Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos que incluya la correlación
- Un cursor abierto en un apodo para una tabla o vista en el origen de datos que incluya la correlación
- Una inserción, supresión o actualización emitida para un apodo de una tabla o vista en el origen de datos que incluya la correlación

Acerca de esta tarea

La sentencia ALTER USER MAPPING se utiliza para modificar el ID de autorización o contraseña empleada en el origen de datos para un ID de autorización del servidor federado específico.

Puede modificar una correlación de usuario desde el Centro de control de DB2 o desde el indicador de línea de mandatos.

Procedimiento

Para modificar una correlación desde el Centro de control de DB2:

1. Expanda la carpeta Objetos de base de datos federada. Los objetos de la correlación de usuario se visualizarán en el panel de contenido de la ventana del Centro de control de DB2.
2. Pulse con el botón derecho del ratón en la correlación de usuario que desee cambiar y, a continuación, pulse **Modificar** en la lista de acciones. Se abrirá la ventana Modificación de correlación de usuario.
3. Cambie el valor de la opción.
4. Pulse **Aceptar** para modificar la correlación de usuario y cierre la ventana Modificar correlación de usuario.

Modificación de una correlación de usuario (línea de mandatos de DB2)

Una correlación de usuario es la asociación entre el ID de autorización del servidor federado y el ID de autorización del origen de datos. Las correlaciones de usuario son necesarias para que las peticiones distribuidas puedan enviarse al origen de datos.

Antes de empezar

Si el ID de autorización que emite la sentencia es diferente al ID de autorización que se correlaciona con el origen de datos, entonces el ID de autorización que emite la sentencia debe incluir la autoridad SYSADM o DBADM en la base de datos federada.

Restricciones

El servidor federado no puede procesar una sentencia ALTER USER MAPPING en una unidad de trabajo determinada (UOW) si dicha UOW ya incluye una de las sentencias siguientes:

- Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos que incluya la correlación
- Un cursor abierto en un apodo para una tabla o vista en el origen de datos que incluya la correlación
- Una inserción, supresión o actualización emitida para un apodo de una tabla o vista en el origen de datos que incluya la correlación

Acerca de esta tarea

La sentencia ALTER USER MAPPING se utiliza para modificar el ID de autorización o contraseña empleada en el origen de datos para un ID de autorización del servidor federado específico.

Puede modificar una correlación de usuario desde el Centro de control de DB2 o desde el indicador de línea de mandatos.

Procedimiento

Para modificar una correlación de usuario desde la línea de mandatos de DB2, emita la sentencia ALTER USER MAPPING:

Por ejemplo, Jenny utiliza el servidor federado para conectarse con un servidor de Sybase denominado SYBSERVER. Accede al servidor federado mediante el ID de autorización *jennifer*. El ID de autorización *jennifer* se correlaciona con el ID de autorización *jenn* del servidor de Sybase. El ID de autorización de Jenny en el servidor de Sybase se ha cambiado por *jen123*. La sentencia ALTER USER MAPPING para correlacionar *jennifer* con *jen123* sería:

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

Tomas utiliza el servidor federado para conectarse con un servidor de Oracle denominado ORASERVER. Accede al servidor federado mediante el ID de autorización *tomas*. El ID de autorización *tomas* se correlaciona con el ID de autorización *tom* del servidor de Oracle. La contraseña de Tomas en el servidor de Oracle ha cambiado. Su nueva contraseña es *day2night*. La sentencia ALTER USER MAPPING para correlacionar *tomas* con la nueva contraseña sería:

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

Las opciones de usuario REMOTE_AUTHID y REMOTE_PASSWORD son sensibles a las mayúsculas y minúsculas, a no ser que se establezcan la opciones de servidor FOLD_ID y FOLD_PW en 'U' o 'L' en la sentencia CREATE SERVER.

Modificación de un apodo (Centro de control de DB2)

Los apodos son identificadores utilizados para hacer referencia a un objeto del origen de datos al que desea acceder. Se pueden cambiar los nombres de las columnas de origen de datos que están almacenadas en el catálogo global y establecer opciones de columna modificando los apodos. Puede modificar el apodo desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Es posible que le interese modificar un apodo para:

- Modificar los nombres de columnas locales para las columnas del objeto de origen de datos
- Modificar los tipos de datos locales para las columnas del objeto de origen de datos
- Añadir, establecer o descartar opciones de columna y de apodo
- Añadir o descartar una clave primaria
- Añadir o descartar una o más restricciones de comprobación, de referencia o exclusivas
- Modificar uno o más de los atributos de las restricciones de dependencia funcional, de comprobación o de referencia
- Evitar el almacenamiento de apodos en la antememoria del servidor federado
- Habilitar el almacenamiento de apodos en la antememoria del servidor federado. Si las tablas de antememoria o las tablas de consultas materializadas están asociadas a un apodo almacenado en la antememoria, debe descartar estas tablas antes de modificar la opción de almacenamiento en antememoria.

Procedimiento

Para modificar un apodo desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**.
2. Pulse con el botón derecho del ratón en el apodo que desee cambiar y, a continuación, pulse **Modificar**. Se abrirá el cuaderno Modificar apodo.
3. En la página Apodos, cambie las opciones aplicables.
4. En la página Claves, defina las restricciones de integridad referencial para el apodo. Puede definir una restricción de clave primaria, de clave exclusiva o de clave foránea.

5. En la página Restricciones de comprobación, defina las restricciones de comprobación o de dependencia funcional para el apodo.
6. En la página Valores, establezca las opciones de apodo para el apodo.
7. Pulse **Aceptar** para modificar el apodo y cierre el cuaderno.

Algunas opciones de apodo son necesarias y no se pueden descartar. Otras opciones de apodo no pueden añadirse si ya se han establecido opciones de apodo específicas.

Cuando el contenido o la estructura del objeto de origen de datos cambien significativamente, debería actualizar las estadísticas de apodo. Entre los cambios significativos se encuentran la adición o eliminación de múltiples filas.

Restricciones en la modificación de apodos

Cuando se modifique un apodo deben tenerse presentes varias restricciones.

Opciones de columna

Si una de las opciones siguientes se ha establecido en una columna, no se podrán añadir más opciones a esa columna:

- SOAPACTIONCOLUMN
- URLCOLUMN
- PRIMARY_KEY
- FOREIGN_KEY

Para BioRS

- Si cambia el nombre de elemento de una columna utilizando la opción `ELEMENT_NAME`, no se comprobará el nuevo nombre para asegurar que es el correcto. Una opción incorrecta puede producir errores cuando se haga referencia a una columna en una consulta.
- Si efectúa cambios en la opción de columna `IS_INDEXED`, estos cambios no los verificará el servidor BioRS. Una opción incorrecta puede producir errores cuando se haga referencia a una columna en una consulta.

Tipos de datos

- Si cambia los tipos de datos de una columna, los nuevos tipos de datos deben ser compatibles con el tipo de datos del elemento o la columna del origen de datos correspondiente. Al cambiar el tipo de datos local por un tipo de datos que sea incompatible con el tipo de datos remoto se pueden producir errores imprevisibles.
- El *local_data_type* no puede ser `LONG VARCHAR`, `LONG VARCHAR`, `XML` o un tipo de datos definidos por el usuario.
- El *data_source_data_type* no puede ser un tipo definido por el usuario.
- Para algunos orígenes de datos no relacionales no se pueden alterar temporalmente los tipos locales existentes o crear tipos locales nuevos. Compruebe la documentación del derivador de origen de datos específico para obtener más información sobre esta restricción.
- Cuando se cambia la especificación local de un tipo de datos de columna, el gestor de bases de datos federadas invalida cualquier estadística (por ejemplo, `HIGH2KEY` y `LOW2KEY`) recopilada para esa columna.

- El tipo local se establece para el objeto de origen de datos específico cuando se accede a este con ese apodo. El mismo objeto de origen de datos puede tener otro apodo que utilice la correlación de tipos de datos por omisión.

Índices

La sentencia ALTER NICKNAME no puede utilizarse para registrar un nuevo índice de origen de datos en la base de datos federada. Utilice la sentencia CREATE INDEX con la cláusula SPECIFICATION para crear una especificación de índice.

Parámetros LOCAL NAME y LOCAL TYPE

- La sentencia ALTER NICKNAME no puede utilizarse para cambiar los nombres locales o tipos de datos para las columnas del apodo si:
 - El apodo se utiliza en una vista, método SQL o función SQL
 - Define una restricción informativa en el apodo
- La cláusula federated_column_options debe especificarse al final, si también necesita especificar el parámetro LOCAL NAME, el parámetro LOCAL TYPE, o ambos, en la sentencia ALTER NICKNAME.

Apodos

La sentencia ALTER NICKNAME no puede utilizarse para cambiar el nombre del banco de datos de BioRS al que hace referencia o se utiliza en un apodo de BioRS. Si el nombre del banco de datos de BioRS cambia, debe descartar el apodo y crearlo de nuevo.

No se puede utilizar la sentencia ALTER NICKNAME para no permitir la colocación en antememoria de un apodo con tablas de antememoria o tablas de consultas materializadas. Debe descartar las tablas de antememoria y las tablas de consultas materializadas antes de no permitir la colocación en antememoria de un apodo.

Unidades de trabajo

El servidor federado no puede procesar una sentencia ALTER NICKNAME en una unidad de trabajo determinada bajo ninguna de las siguientes condiciones:

- Si el apodo al que se hace referencia en la sentencia ALTER NICKNAME tiene un cursor abierto en esta en la misma unidad de trabajo.
- Si se emite una inserción, supresión o actualización en la misma unidad de trabajo para el apodo al que se hace referencia en la sentencia ALTER NICKNAME.

Modificación de nombres de columnas de apodo (Centro de control de DB2)

Se puede modificar un apodo para cambiar los nombres de columnas. Puede modificar los nombres de las columnas desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia

- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Cuando se crea un apodo, los nombres de columnas que están asociados al objeto de origen de datos se almacenan en la base de datos federada. Para algunos orígenes de datos, el derivador especifica los nombres de columnas para el usuario. Para otros orígenes de datos, el usuario deberá especificar los nombres de columnas cuando cree el apodo.

Procedimiento

Para modificar los nombres de columnas de apodo desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**.
2. Pulse con el botón derecho del ratón en el apodo que desee cambiar y, a continuación, pulse **Modificar**. Se abrirá el cuaderno Modificar apodo.
3. En la página Apodos, seleccione la columna que desee cambiar y pulse **Cambiar**. Se abrirá la ventana Cambiar columna.
4. Escriba el nombre de la columna.
5. Pulse en **Aceptar** para cambiar nombre de la columna y cierre la ventana.
6. Pulse **Aceptar** para modificar el apodo y cierre el cuaderno.

Modificación de nombres de columnas de apodo (línea de mandatos de DB2)

Se puede modificar un apodo para cambiar los nombres de columnas. Puede modificar los nombres de las columnas desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Cuando se crea un apodo, los nombres de columnas que están asociados al objeto de origen de datos se almacenan en la base de datos federada. Para algunos orígenes de datos, el derivador especifica los nombres de columnas para el usuario. Para otros orígenes de datos, el usuario deberá especificar los nombres de columnas cuando cree el apodo.

Procedimiento

Para modificar los nombres de columnas de apodo desde la línea de mandatos de DB2, emita la sentencia ALTER NICKNAME.

```
ALTER NICKNAME nickname
ALTER COLUMN current_name
LOCAL NAME new_name
```

Modificación de opciones de apodo (Centro de control de DB2)

Las opciones de apodo son parámetros que se especifican en un apodo cuando se emiten las sentencias CREATE NICKNAME y ALTER NICKNAME. Se pueden añadir, establecer o descartar opciones de apodo utilizando la sentencia ALTER NICKNAME. Puede modificar los nombres de las columnas desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Por ejemplo, se ha creado el apodo DRUGDATA1 para el archivo con estructura de tabla drugdata1.txt. La vía de acceso totalmente calificada definida originalmente en la sentencia CREATE NICKNAME era /user/pat/drugdata1.txt.

Para modificar la opción de apodo FILE_PATH, emita la sentencia siguiente:

Procedimiento

Para cambiar los nombres de columnas desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**.
2. Pulse con el botón derecho del ratón en el apodo que desee cambiar y, a continuación, pulse **Modificar**. Se abrirá el cuaderno Modificar apodo.

3. En la página de Valores, seleccione el recuadro de selección situado junto a cada opción que desee añadir o eliminar. No se pueden eliminar las opciones necesarias.
4. Para especificar o cambiar el valor de una opción, pulse en el campo **Valor** para dicha opción. Dependiendo de la opción, puede seleccionar un valor de una lista, pulsar para seleccionar múltiples valores o escribir el nuevo valor.
5. Pulse **Aceptar** para modificar el apodo y cierre el cuaderno.

Modificación de opciones de apodo (línea de mandatos de DB2)

Las opciones de apodo son parámetros que se especifican en un apodo cuando se emiten las sentencias CREATE NICKNAME y ALTER NICKNAME. Se pueden añadir, establecer o descartar opciones de apodo utilizando la sentencia ALTER NICKNAME. Puede modificar las opciones de apodo desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Procedimiento

Para modificar las opciones de apodo desde el indicador de línea de mandatos, emita la sentencia ALTER NICKNAME.

```
ALTER NICKNAME nickname
    OPTIONS (SET option_name 'option_string_value')
```

Ejemplo: Se ha creado el apodo DRUGDATA1 para el archivo con estructura de tabla drugdata1.txt. La vía de acceso totalmente calificada definida originalmente en la sentencia CREATE NICKNAME era /user/pat/drugdata1.txt. Para modificar la opción de apodo FILE_PATH, emita la sentencia siguiente:

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

Modificación de opciones de columnas de apodo (Centro de control de DB2)

Se pueden añadir, establecer o descartar opciones de columnas de apodo utilizando la sentencia ALTER NICKNAME. Puede modificar los nombres de las columnas desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Debe especificar información sobre columnas en las sentencias CREATE NICKNAME y ALTER NICKNAME utilizando los parámetros denominados opciones de columnas de apodo.

Procedimiento

Para modificar las opciones de columnas de apodo desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**.
2. Pulse con el botón derecho del ratón en el apodo que desee cambiar y, a continuación, pulse **Modificar**. Se abrirá el cuaderno Modificar apodo.
3. En la página Apodos, seleccione la columna que desee cambiar y pulse **Cambiar**. Se abrirá la ventana Cambiar columna.
4. Seleccione la opción de columna que desee añadir o eliminar.
5. Para las opciones que añada o cambie, especifique el valor de una opción.
6. Pulse en **Aceptar** para cambiar la opción de columna y cierre la ventana.
7. Pulse **Aceptar** para modificar el apodo y cierre el cuaderno.

Modificación de opciones de columnas de apodo (línea de mandatos de DB2)

Se pueden añadir, establecer o descartar opciones de columnas de apodo utilizando la sentencia ALTER NICKNAME. Puede modificar los nombres de las columnas desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Debe especificar información sobre columnas en las sentencias CREATE NICKNAME y ALTER NICKNAME utilizando los parámetros denominados opciones de columnas de apodo. Puede especificar cualquiera de estos valores en letras mayúsculas o minúsculas.

Procedimiento

Para modificar las opciones de columnas de apodo desde el indicador de línea de mandatos, utilice la sentencia ALTER NICKNAME.

Ejemplo 1: Especificación de la opción de columna NUMERIC_STRING con orígenes de datos relacionales

La opción de columna NUMERIC_STRING se aplica a columnas de tipo carácter (CHAR y VARCHAR). Suponga que un origen de datos tiene una secuencia de clasificación que difiere de la secuencia de clasificación de la base de datos federada. Normalmente, el servidor federado no clasificaría ninguna columna que contuviese datos de carácter en el origen de datos. Devolvería los datos a la base de datos federada y realizaría la clasificación localmente. Sin embargo, suponga que los datos de la columna son de tipo carácter y que la columna sólo contiene caracteres numéricos ('0','1',..., '9'). Puede indicar este hecho asignando un valor 'Y' a la opción de columna NUMERIC_STRING. Esto proporciona al optimizador de consultas DB2 UDB la posibilidad de realizar la clasificación en el origen de datos. Si la clasificación se realiza de forma remota, puede evitar la actividad general que supone clasificar los datos en el servidor federado.

El apodo ORA_INDSALES es para una tabla de Oracle denominada INDONESIA_SALES. La tabla contiene la columna POSTAL_CODE con el tipo de datos VARCHAR. Originalmente, la columna contenía solamente caracteres numéricos, y la opción de columna NUMERIC_STRING se estableció en 'Y'. No obstante, ahora la columna contiene una mezcla de caracteres numéricos y no numéricos. Para modificar la opción de columna NUMERIC_STRING a 'N', utilice esta sentencia:

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

Ejemplo 2: Especificación de la opción de columna VARCHAR_NO_TRAILING_BLANKS con orígenes de datos relacionales

La opción de columna VARCHAR_NO_TRAILING_BLANKS puede utilizarse para identificar columnas específicas que no contengan blancos de cola. El compilador de SQL tendrá en cuenta este valor cuando compruebe todas las operaciones (por ejemplo, operaciones de comparación) realizadas en las columnas.

El apodo ORA_INDSALES es para una tabla de Oracle denominada INDONESIA_SALES. La tabla contiene la columna NAME_CODE con el tipo de datos VARCHAR. La columna NAME no contiene blancos de cola. Para añadir la opción VARCHAR_NO_TRAILING_BLANKS al apodo, utilice esta sentencia:

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


Ejemplo 3: Especificación de la opción de columna XPATH con orígenes de datos no relacionales

El apodo EMPLOYEE es para un origen de datos XML. Se ha especifica la opción XPATH para la columna *fname*. Para establecer la opción de columna XPATH en una vía de acceso diferente, utilice esta sentencia:

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

Modificación de un apodo (línea de mandatos de DB2)

Los apodos son identificadores utilizados para hacer referencia a un objeto del origen de datos al que desea acceder. Se pueden cambiar los nombres de las columnas de origen de datos que están almacenadas en el catálogo global y establecer opciones de columna modificando los apodos. Puede modificar el apodo desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Es posible que le interese modificar un apodo para:

- Modificar los nombres de columnas locales para las columnas del objeto de origen de datos
- Modificar los tipos de datos locales para las columnas del objeto de origen de datos
- Añadir, establecer o descartar opciones de columna y de apodo
- Añadir o descartar una clave primaria
- Añadir o descartar una o más restricciones de comprobación, de referencia o exclusivas
- Modificar uno o más de los atributos de las restricciones de dependencia funcional, de comprobación o de referencia

Procedimiento

Para modificar un apodo desde la línea de mandatos de DB2, emita la sentencia ALTER NICKNAME con los parámetros apropiados establecidos.

Cuando el contenido o la estructura del objeto de origen de datos cambien significativamente, debería actualizar las estadísticas de apodo. Entre los cambios significativos se encuentran la adición o eliminación de múltiples filas.

Descarte de un derivador

Existen varias razones por las que podría desear descartar un derivador.

Antes de empezar

Pare emitir la sentencia DROP WRAPPER, debe tener autoridad SYSADM o DBADM.

Acerca de esta tarea

En ocasiones existe más de un derivador que puede utilizar para acceder al origen de datos. El hecho de escoger uno de ellos puede depender de la versión de software de cliente de origen de datos que utilice. O puede depender del sistema operativo que utilice en su servidor federado. Suponga que desea acceder a dos tablas de Oracle y una vista de Oracle. Está utilizando Oracle Versión 10 y el sistema operativo en su servidor federado es Windows. Originalmente, había creado el derivador SQLNET. Dado que IBM InfoSphere Federation Server no da soporte al derivador SQLNET, puede descartar dicho derivador SQLNET y crear el derivador NET8.

Otra razón para descartar un derivador sería que ya no necesite acceder al origen de datos a la que el derivador está asociado. Por ejemplo, suponga que su organización tiene un requisito para acceder a la información sobre clientes tanto en bases de datos de servidores de Informix y Microsoft SQL. Crea un derivador para el origen de datos de Informix y un derivador para el origen de datos de Microsoft SQL. Después, su organización decide migrar toda la información desde Microsoft SQL Server a Informix. Ya no necesita el derivador de Microsoft SQL Server y puede descartarlo.

Importante: Descartar un derivador puede tener graves consecuencias. Otros objetos que había registrado en el servidor federado pueden verse afectados:

- Todas las definiciones de servidor que sean dependientes del derivador descartado también serán descartadas.
- Todos los objetos que sean dependientes de las definiciones de servidor descartadas también serán descartados.
- Todos los apodos que sean dependientes de las definiciones de servidor descartadas también serán descartados. Al descartar los apodos dependientes de la definición de servidor los objetos dependientes de esos apodos se verán afectados:
 - Cualquier especificación de índice dependiente de los apodos descartados también será descartada.
 - Cualquier vista dependiente de los apodos descartados será marcada como no operativa.
 - Cualquier tabla de consultas materializadas dependiente de los apodos descartados también será descartada.
- Todos los paquetes y sentencias de SQL dinámico almacenadas en antememoria dependientes de los apodos descartados serán marcados como inválidos y permanecerán así hasta que se vuelvan a crear los objetos dependientes.

Procedimiento

Para descartar un derivador, utilice la sentencia DROP.

Ejemplo: Descartar el derivador *MSSQLODBC3* de Microsoft SQL Server:

```
DROP WRAPPER MSSQLODBC3
```

Descarte de una definición de servidor

Al descartar una definición de servidor se suprime la definición del catálogo global. El objeto de origen de datos al que hace referencia la definición de servidor no se verá afectado. Puede descartar una definición de servidor utilizando el Centro de control de DB2 o utilizando la sentencia DROP desde el procesador de línea de mandatos de DB2.

Antes de empezar

Para descartar una definición de servidor debe tener autoridad SYSADM o DBADM.

Restricciones

El servidor federado no puede procesar una sentencia DROP SERVER en una unidad de trabajo (UOW) determinada bajo ninguna de las siguientes condiciones:

- La sentencia hace referencia a una sola origen de datos, y la UOW ya incluye una de las sentencias siguientes:
 - Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en el origen de datos
 - Un cursor abierto en un apodo para una tabla o vista en el origen de datos
 - Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en el origen de datos
- La sentencia hace referencia a una categoría de orígenes de datos (por ejemplo, todas los orígenes de datos de un tipo y versión específicos), y la UOW ya incluye una de las sentencias siguientes:
 - Una sentencia SELECT que haga referencia a un apodo para una tabla o vista en una de los orígenes de datos
 - Un cursor abierto en un apodo para una tabla o vista en una de los orígenes de datos
 - Una inserción, supresión o actualización emitida contra un apodo de una tabla o vista en una de los orígenes de datos

Acerca de esta tarea

Cuando ya no necesite acceder a un servidor de orígenes de datos, descarte la definición de servidor de la base de datos federada. Cuando descarte una definición de servidor, otros objetos que había registrado en el servidor federado pueden verse afectados:

- Todas las correlaciones de funciones definidas por el usuario, las correlaciones de tipos de datos definidos por el usuario y las correlaciones de usuarios que sean dependientes de la definición de servidor también serán descartadas.
- Todos los apodos que sean dependientes de la definición de servidor descartada también serán descartados. Al descartar los apodos dependientes de la definición de servidor los objetos dependientes de esos apodos se verán afectados:

- Cualquier especificación de índice dependiente de los apodos descartados también será descartada.
- Cualquier vista dependiente de los apodos descartados será marcada como no operativa.
- Cualquier tabla de consultas materializadas dependiente de los apodos descartados también será descartada.
- Todos los paquetes y sentencias de SQL dinámico almacenadas en antememoria dependientes de los apodos descartados serán marcados como inválidos y permanecerán así hasta que se vuelvan a crear los objetos dependientes.

Procedimiento

Para suprimir una definición de servidor, emita la sentencia DROP:

```
DROP SERVER server_name
```

dónde *server_name* identifica la definición de servidor que debe descartarse.

Ejemplo: Un servidor de Informix utiliza el nombre de servidor INFMX01. La sentencia DROP siguiente descarta la definición de servidor:

```
DROP SERVER INFMX01
```

Descarte de una correlación de usuario

Cuando un usuario ya no necesite acceder a un origen de datos remota, descarte la correlación de usuario entre el servidor federado y el servidor de origen de datos remoto. Si el usuario está correlacionado con más de un servidor de orígenes de datos, deberá descartar cada correlación por separado.

Antes de empezar

Para emitir una sentencia DROP USER MAPPING, el ID de autorización de la sentencia DROP debe tener autoridad SYSADM o DBADM, si dicho ID de autorización es distinto del ID de usuario de la base de datos federada especificado en la correlación de usuario. En caso contrario, si el ID de autorización y el ID de usuario de la correlación de usuario coinciden, no se necesitan ni privilegios ni autoridades.

Procedimiento

Para suprimir una correlación de usuario, emita la sentencia DROP:

```
DROP USER MAPPING FOR user_ID SERVER local_server_name
```

donde:

- *user_ID* es el ID de autorización para el usuario en el servidor federado
- *local_server_name* es el nombre local utilizado para identificar el servidor de orígenes de datos remoto en la definición de servidor.

Descarte de un apodo

Al descartar un apodo se suprime el apodo del catálogo global en el servidor federado. El objeto de origen de datos al que hace referencia el apodo no se verá afectado.

Antes de empezar

El apodo debe listarse en el catálogo.

El ID de autorización de la sentencia DROP cuando se descartan apodos debe tener al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio DROPIN en el esquema para el apodo
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.
- Privilegio CONTROL en el apodo

Restricciones

Para los apodos para orígenes de datos no relacionales, el servidor federado no puede procesar una sentencia DROP NICKNAME en una unidad de trabajo (UOW) determinada bajo ninguna de las siguientes condiciones:

- Si el apodo al que se hace referencia en la sentencia tiene un cursor abierto en esta en la misma UOW.
- Si el apodo al que se hace referencia en esta sentencia ya está referenciado por una sentencia SELECT en la misma UOW.
- Si se emite una inserción, supresión o actualización en la misma UOW para el apodo al que se hace referencia en la sentencia.

Para los apodos para orígenes de datos no relacionales, el servidor federado no puede procesar una sentencia DROP NICKNAME en una unidad de trabajo (UOW) determinada bajo ninguna de las siguientes condiciones:

- Si el apodo al que se hace referencia en la sentencia tiene un cursor abierto en esta en la misma UOW.
- Si el apodo al que se hace referencia en esta sentencia ya está referenciado por una sentencia SELECT en la misma UOW.

Acerca de esta tarea

Cuando descarte un apodo, otros objetos que había registrado en el servidor federado pueden verse afectados:

- Descartar los apodos afectará a los objetos dependientes de esos apodos:
 - Cualquier especificación de índice dependiente de los apodos descartados también será descartada.
 - Cualquier vista dependiente de los apodos descartados será marcada como no operativa.
 - Cualquier tabla de consultas materializadas dependiente de los apodos descartados también será descartada.
- Todos los paquetes y sentencias de SQL dinámico almacenadas en antememoria dependientes del apodo descartado serán marcados como inválidos y permanecerán así hasta que se vuelvan a crear los objetos dependientes.

Procedimiento

Para suprimir un apodo, emita la sentencia DROP:

```
DROP NICKNAME nickname
```

dónde *nickname* identifica el apodo que debe descartarse.

Capítulo 3. Correlaciones de tipos de datos en un sistema federado

Los tipos de datos en un origen de datos deben correlacionarse con los tipos de datos de DB2 correspondientes. Dicha correlación permite que el servidor federado recupere los datos del origen de datos.

La base de datos federada proporciona un conjunto de correlaciones de tipos de datos por omisión para algunos orígenes de datos. Para otros orígenes de datos el usuario deberá proporcionar las correlaciones de tipos de datos que desee utilizar. Para los orígenes de datos no relacionales, no se podrán alterar temporalmente las correlaciones de tipos de datos existentes ni crear nuevas correlaciones.

A continuación se muestran algunos ejemplos de correlaciones de tipos de datos por omisión:

- El tipo FLOAT de Oracle se correlaciona por omisión con el tipo DOUBLE de DB2
- El tipo DATE de Oracle se correlaciona por omisión con el tipo TIMESTAMP de DB2
- El tipo DATE de DB2 para z/OS se correlaciona por omisión con el tipo DATE de DB2

Los apodos que se creen después de que se haya cambiado una correlación utilizarán la correlación de tipos de datos nueva. Los apodos que se creen antes de que se haya cambiado una correlación utilizarán la correlación de tipos de datos por omisión.

Si ya ha creado los apodos, podrá actualizar los apodos existentes de una de las maneras siguientes:

- Puede alterar cada apodo
- Puede descartar y volver a crear cada apodo

Los servidores federados de DB2 no dan soporte a las correlaciones para los tipos de datos siguientes:

- El tipo de datos local no puede ser LONG VARCHAR, LONG VARGRAPHIC o un tipo de datos definido por el usuario.
- El tipo de datos remoto no puede ser un tipo definido por el usuario.

No obstante, puede utilizar una función de conversión para convertir el tipo de datos definido por el usuario en una vista del origen de datos en un tipo de datos de sistemas o incorporados. A continuación, puede crear un apodo para la vista. Para la mayoría de orígenes de datos, si crea tales vistas, éstas no contendrán estadísticas ni índices y no podrán actualizarse.

Correlaciones de tipos de datos y el catálogo global de bases de datos federadas

Las definiciones de tipos de datos locales se almacenan en la vista de catálogo SYSCAT.COLUMNS del catálogo global de bases de datos federadas.

Cuando grabe una sentencia `CREATE NICKNAME`, especifique un objeto de origen de datos al que represente el apodo. En la mayoría de los casos, el servidor federado define un tipo de datos soportado por DB2 para cada columna o campo de dicho objeto de origen de datos. Para algunos orígenes de datos no relacionales, el usuario deberá proporcionar el tipo de datos de DB2.

Para los orígenes de datos relacionales, con el fin de determinar qué tipo de datos local deben almacenarse en la vista de catálogo `SYSCAT.COLUMNS`, el servidor federado busca información de correlaciones de tipos de datos directas en los derivadores y en la vista de catálogo `SYSCAT.TYPEMAPPINGS`. Las correlaciones en la vista de catálogo `SYSCAT.TYPEMAPPINGS` tienen prioridad sobre las correlaciones por omisión en los derivadores. Si se crean correlaciones alternativas para alterar temporalmente las correlaciones de tipos de datos por omisión, el servidor federado utilizará dichas correlaciones alternativas. Cuando múltiples correlaciones se pueden aplicar a una columna, el servidor federado utilizará las correlaciones creadas más recientemente.

Para los orígenes de datos no relacionales, con el fin de determinar qué tipo de datos local deben almacenarse en la vista de catálogo `SYSCAT.COLUMNS`, el servidor federado busca información de correlaciones de tipos de datos en los derivadores. Dependiendo del origen de datos no relacional, variará el grado en que puede modificar los tipos de datos definidos por el desviador. Para algunos orígenes no relacionales, no debe especificar ninguna columna. El derivador define los tipos de datos. Para otros tipos de orígenes de datos, puede alterar temporalmente los tipos de datos. Y para otros orígenes de datos, debe especificar los tipos de datos de columna en la sentencia `CREATE NICKNAME`.

Cuando grabe el DDL transparente de `CREATE TABLE` para orígenes de datos relacionales, especifique los tipos de datos de DB2 en la sentencia. El servidor federado busca información sobre las correlaciones de tipos de datos inversas entre la base de datos federada y el origen de datos. El servidor federado busca dicha información en la vista de catálogo `SYSCAT.TYPEMAPPINGS`.

Cuando los valores de una columna de origen de datos se devuelven a la base de datos federada, los valores se ajustarán totalmente al tipo de datos de DB2 con el que se correlaciona la columna de origen de datos. Si dicha correlación es por omisión, los valores también se ajustarán totalmente con el tipo de origen de datos en la correlación. Por ejemplo, si una tabla de Oracle con una columna `FLOAT` se define para una base de datos federada, la correlación por omisión de `FLOAT` con `DOUBLE` de DB2 se aplica automáticamente a dicha columna. Los valores que se devuelven desde la columna se ajustarán totalmente tanto a los tipos de datos `DOUBLE` como `FLOAT`.

Cuándo se deben crear correlaciones de tipos de datos alternativas

Se pueden crear correlaciones de tipos de datos alternativas para orígenes de datos relacionales.

Puede que desee crear correlaciones de tipos de datos alternativas en las situaciones siguientes:

- Para alterar temporalmente una correlación de tipos de datos por omisión
Para algunos derivadores, puede cambiar el formato o longitud de los valores que se devuelven. Podrá modificar el formato o la longitud cambiando el tipo de datos de DB2 a los que se deben ajustar los valores. Por ejemplo, el tipo de datos `DATE` de Oracle se utiliza como una indicación de fecha y consta de siglo,

año, mes, día, hora, minutos y segundos. Por omisión, el tipo de datos DATE de Oracle se correlaciona con el tipo de datos TIMESTAMP de DB2. Para que devuelva solamente la información sobre la hora, minutos y segundos, puede alterar temporalmente la correlación de tipos de datos por omisión, de forma que el tipo de datos DATE de Oracle se correlacione con el tipo de datos TIME de DB2. Cuando se consultan las columnas DATE de Oracle, sólo la porción horaria de los valores de indicación de fecha de Oracle se devuelven al servidor federado.

- Cuándo no existe una correlación por omisión
Cuando la correlación de tipos de datos no esté disponible para un tipo de datos del origen de datos, deberá crear una correlación para el nuevo tipo de datos.

Debe utilizar la sentencia CREATE TYPE MAPPING para crear nuevas correlaciones de tipos de datos. Las correlaciones que cree se almacenarán en la vista de catálogo SYSCAT.TYPEMAPPINGS de la base de datos federada.

Correlaciones de tipos de datos para orígenes de datos no relacionales

Para algunos orígenes de datos no relacionales, las correlaciones de tipos de datos no se encuentran en los derivadores. En algunos casos, se debe especificar la información de tipo local en la sentencia CREATE NICKNAME.

El ejemplo siguiente muestra cómo se especifican los tipos de datos de columna en la sentencia CREATE NICKNAME para algunas bases de datos no relacionales:

```
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')
```

Correlaciones de tipos de datos directas e inversas

Las correlaciones de tipos de datos directas e inversas son dos tipos de correlaciones entre los tipos de datos del origen de datos y los tipos de datos de la base de datos federada.

Una *correlación de tipos de datos directa* es la correlación de un tipo de datos remoto con un tipo de datos local comparable. Las correlaciones de tipos de datos directas se utilizan cuando se crea un apodo para un objeto de origen de datos. El tipo local comparable para cada columna del objeto de origen de datos se almacena en el catálogo global.

Una *correlación de tipos de datos inversa* es la correlación de un tipo de datos local con un tipo de datos remoto comparable. La correlación de tipos de datos inversa se utiliza con DDL transparente.

Figura 2 en la página 50 muestra la correlación de tipos de datos directa e inversa.

Correlación de tipo de datos

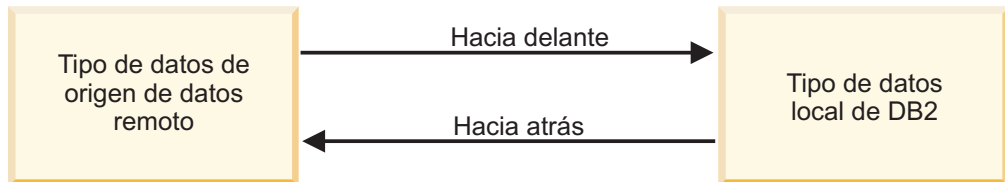


Figura 2. Correlaciones de tipos de datos directas e inversas

Creación de correlaciones de tipos de datos

Utilice la sentencia `CREATE TYPE MAPPING` para crear una correlación de tipo de datos. Puede ejecutar la sentencia desde el Centro de control de DB2 o desde el procesador de línea de mandatos o incluirla en un programa de aplicaciones. No puede utilizar el Centro de control de DB2 para crear o modificar correlaciones de tipos de datos.

Antes de empezar

Los privilegios contenidos en el ID de autorización asociado con la sentencia deben tener autoridad `SYSADM` o `DBADM`.

Acerca de esta tarea

Consulte el apartado Sentencia `CREATE TYPE MAPPING` para obtener información específica de uso.

Restricciones

- El valor del `local_data_type` no puede ser `LONG VARCHAR`, `LONG VARGRAPHIC` o un tipo de datos definidos por el usuario.
- El valor del `data_source_data_type` no puede ser un tipo definido por el usuario.
- Para orígenes de datos no relacionales, el grado en que se pueden alterar temporalmente las correlaciones de tipos de datos existentes o crear correlaciones es limitado.

Procedimiento

Ejecute la sentencia `CREATE TYPE MAPPING` para crear una correlación de tipo de datos.

Para especificar un tipo de servidor en la sentencia `CREATE TYPE MAPPING`, debe utilizar uno de los siguientes valores tales como `server-type`:

Tabla 4. Tipos de servidores válidos

Origen de datos	Tipo de servidor
DB2 Database para Linux, UNIX y Windows	DB2/CS DB2/UDDB2/NT DB2/SUN DB2/HP DB2/HPUX DB2/AIX DB2/6000 DB2/PE DB2/PTX DB2/SCO DB2/LINUX DB2/EEE DB2/2
DB2 para z/OS	DB2/MVS DB2/ZOSDB2/390
DB2 Server para VSE y VM	DB2/VMDB2/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

Creación de una correlación de tipos para un tipo de origen de datos - ejemplo

En este ejemplo, todas las tablas y vistas de Oracle que utilizan el tipo de datos NUMBER de Oracle deben correlacionarse con el tipo de datos DECIMAL (8,2) de DB2. El tipo de datos NUMBER de Oracle se correlaciona por omisión con el tipo de datos DOUBLE de DB2, un tipo de datos decimales de coma flotante.

Utilice la sentencia ALTER NICKNAME para modificar los tipos locales de los apodos existentes. Debe modificar cada apodo por separado para cambiar el tipo de datos local a DECIMAL (8,2). Si los apodos no existen, cree una correlación de tipo de datos que especifique el tipo de origen de datos. Asegúrese de que se cree el derivador de origen de datos antes de iniciar la sentencia CREATE TYPE MAPPING. Para crear la correlación de tipos desde el tipo de datos NUMBER de Oracle en el tipo de datos DECIMAL(8,2) de DB2, ejecute la sentencia CREATE TYPE MAPPING, por ejemplo:

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

MY_ORACLE_DEC

Nombre que se ha dado a la correlación de tipos. El nombre no puede duplicar un nombre de correlación de tipo de datos que ya exista en el catálogo.

FROM *SYSIBM.DECIMAL(8,2)*

El esquema de DB2 local y el tipo de datos local. Si no se especifica la longitud o la precisión y la escala, entonces dichos valores se determinan desde el tipo de datos del origen.

TO SERVER TYPE *ORACLE*

El tipo de origen de datos.

TYPE *NUMBER*

El tipo de datos de origen de datos que se correlaciona con el tipo de datos local. No se permiten los tipos de datos definidos por el usuario.

El tipo de datos DECIMAL (8,2) DB2 se define de manera local para las columnas de Oracle. Cuando se crean apodos en las tablas y vistas de Oracle que contienen columnas NUMBER, el tipo de datos NUMBER de Oracle se correlaciona con el tipo de datos DECIMAL (8,2) de DB2.

Creación de una correlación de tipos para un tipo de origen de datos y versión - ejemplo

En este ejemplo, las tablas y vistas de Oracle existen en diferentes versiones del servidor de Oracle. Para todas las tablas y vistas en servidores de Oracle Versión 8.0.3, las columnas que utilizan el tipo de datos NUMBER (23,3) de Oracle deben correlacionarse con el tipo de datos DECIMAL (8,2) de DB2.

El tipo de datos NUMBER (23,3) de Oracle se correlaciona por omisión con el tipo de datos DECIMAL (23,3) de DB2. Utilice la sentencia ALTER NICKNAME para modificar los tipos locales de los apodos existentes. Debe modificar cada apodo por separado para cambiar el tipo de datos local a DECIMAL (8,2). Si los apodos no existen, cree una correlación de tipo de datos que especifique el tipo de origen de datos. Asegúrese de que se cree el derivador de origen de datos antes de iniciar la sentencia CREATE TYPE MAPPING. Para correlaciones el tipo de datos NUMBER(23,3) de Oracle con el tipo de datos DECIMAL(8,2) de DB2 para servidores Oracle utilizando la versión 8.0.3, ejecute la sentencia CREATE TYPE MAPPING, por ejemplo:

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

ORA_DEC

Nombre que se ha dado a la correlación de tipos. El nombre no puede duplicar un nombre de correlación de tipo de datos que ya exista en el catálogo.

FROM *SYSIBM.DECIMAL(8,2)*

El esquema de DB2 local y el tipo de datos local. Si no se especifica la longitud o la precisión y la escala, entonces dichos valores se determinan desde el tipo de datos del origen.

TO SERVER TYPE *ORACLE*

El tipo de origen de datos.

VERSION *8.0.3*

La versión del servidor de orígenes de datos. Debe especificar la versión. También debe especificar el release y la modificación del release, tal como se muestra en este ejemplo.

TYPE *NUMBER(23,3)*

El tipo de datos de origen de datos que se correlaciona con el tipo de datos local. No se permiten los tipos de datos definidos por el usuario.

La base de datos federada define el tipo de datos DECIMAL (8,2) de DB2 de forma local para las columnas de Oracle en los servidores de la Versión 8.0.3. Las tablas y vistas en servidores que no utilizan la Versión 8.0.3 emplean la correlación de tipo de datos por omisión. Cuando se crean apodos en las tablas y vistas de Oracle que contienen columnas NUMBER, el tipo de datos NUMBER de Oracle se correlaciona con el tipo de datos DECIMAL (8,2) de DB2.

Creación de una correlación de tipos para todos los objetos de origen de datos en un servidor - ejemplo

En este ejemplo, el servidor se define para la base de datos federada como ORA2SERVER. Cada tabla contiene una columna con un tipo de datos DATE de Oracle.

El tipo de datos DATE de Oracle consta de siglo, año, mes, día, hora, minutos y segundos. El tipo de datos DATE de Oracle se correlaciona por omisión con el tipo de datos TIMESTAMP de DB2 local. No obstante, cuando se consulte cualquier objeto de este servidor, el conjunto de resultados debe devolver sólo la información horaria (hora, minutos y segundos).

Utilice la sentencia ALTER NICKNAME para modificar los tipos locales de los apodos existentes. Debe modificar cada apodo por separado para cambiar el tipo de datos local a TIME.

Si los apodos no existen, cree una correlación de tipo de datos que especifique el tipo de origen de datos.

Para correlacionar el tipo de datos DATE de Oracle con el tipo de datos TIME de DB2 para el ORA2SERVER, emita la sentencia siguiente:

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

ORA2_DATE

Nombre que se ha dado a la correlación de tipos. El nombre no puede duplicar un nombre de correlación de tipo de datos que ya exista en el catálogo.

FROM SYSIBM.TIME

El esquema de DB2 local y el tipo de datos local. Si no se especifica la longitud o la precisión y la escala, entonces dichos valores se determinan desde el tipo de datos del origen.

TO SERVER ORA2SERVER

El nombre local del servidor de orígenes de datos.

TYPE DATE

El tipo de datos de origen de datos que se correlaciona con el tipo de datos local. No se permiten los tipos de datos definidos por el usuario.

La base de datos federada define localmente el tipo de datos TIME de DB2 para las columnas de Oracle de tipo de datos DATE.

Cuando se crean apodos en las tablas y vistas de Oracle que contienen columnas DATE, el tipo de datos DATE de Oracle se correlaciona con el tipo de datos DECIMAL (8,2) de DB2 .

Los objetos de origen de datos en otros servidores de Oracle no se ven afectados por esta correlación de tipos de datos.

Conversión entre tipos de datos

Puede convertir un tipo de datos de tipo de datos de origen a tipo de datos de destino.

La conversión entre tipos de datos se pueden producir de modo implícito o de modo explícito.

- La conversión implícita es una conversión automática de datos de un tipo de datos a otro tipo de datos basado en un conjunto tácito de reglas de conversión. Este conversión automática se produce con ayuda de la programación dinámica (weak typing).
- La conversión explícita soporta la programación estática (strong typing). La programación estática requiere que los tipos de datos sean coincidentes. Debe convertir explícitamente uno o ambos tipos de datos en un tipo de datos común para poder realizar comparaciones o asignaciones.

E soporte de Federation para la conversión de tipos entre tipos de datos permite a las consultas federadas sobre apodos acceder a los servidores que soportan tanto la programación dinámica como la programación estática.

Las funciones de la conversión de tipos no se puede forzar en los servidores remotos si no existe una función equivalente en el servidor remoto. Si utiliza con exceso la conversión de tipos puede provocar problemas de rendimiento.

Ejemplos: conversión de tipos explícita

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

Ejemplos: conversión de tipos implícita

Ejemplo 1:

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

En esta operación de asignación, la sentencia que se ha enviado al servidor remoto es equivalente a la sentencia siguiente:

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

Ejemplo 2:

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

En esta operación de asignación, la sentencia que se ha enviado al servidor remoto es equivalente a la sentencia siguiente:

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

Ejemplo 3:

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

En esta operación de comparación, la sentencia que se ha enviado al servidor remoto es equivalente a la sentencia siguiente:

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

Soporte para tipos de datos **TIMESTAMP**

El tipo de datos **TIMESTAMP** está parametrizado para que controle la precisión de segundos fraccionales.

Para orígenes de datos de DB2 para Linux, UNIX y Windows, las indicaciones de fecha y hora se correlacionan con **TIMESTAMP(*p*)**, donde *p* representa la precisión y especifica el número de segundos fraccionales. El rango de *p* es de 0 a 12, incluido.

Para otros orígenes de datos, las indicaciones de fecha y hora se correlacionan con **TIMESTAMP** con una precisión predeterminada de 6. Para estos orígenes de datos, puede aprovechar el soporte de **TIMESTAMP(*p*)** utilizando correlaciones similares a los ejemplos proporcionados para correlaciones de tipos predeterminadas.

Cuando una columna **TIMESTAMP** de apodo tiene una precisión más pequeña que la columna de la tabla remota correspondiente, y la columna de la tabla remota contiene dígitos que no sean un cero en los dígitos extra fraccionales, el resultado de predicados que utilizan la columna de apodo puede ser imprevisible. El resultado de los predicados que utilizan esa columna de apodo trabaja correctamente si el predicado no se elimina.

Modificación de un tipo local para un objeto de origen de datos (Centro de control de DB2)

Para modificar un tipo de datos local debe utilizar la sentencia **ALTER NICKNAME** en lugar de la sentencia **CREATE TYPE MAPPING**. Puede modificar el tipo de datos desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad **SYSADM** o **DBADM**
- Privilegio **ALTER** en el apodo especificado en la sentencia
- Privilegio **CONTROL** en el apodo especificado en la sentencia
- Privilegio **ALTERIN** en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna **DEFINER** de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Cuando se crea un apodo, los tipos de datos que están asociados al objeto de origen de datos se almacenan en la base de datos federada. Para algunos orígenes de datos, el derivador especifica los tipos de datos para el usuario. Para otros orígenes de datos, deberá especificar los tipos de datos cuando cree el apodo.

Puede especificar un tipo local para una columna de un objeto de origen de datos específico.

Importante: cambiar el tipo de datos local puede resultar en errores o pérdida de información si se modifica el tipo de datos local para una columna por un tipo que difiera significativamente de su tipo remoto.

Procedimiento

Para cambiar el tipo de datos local desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**.
2. Pulse con el botón derecho del ratón en el apodo que desee cambiar y, a continuación, pulse **Modificar**. Se abrirá el cuaderno Modificar apodo.
3. En la página Apodos, seleccione la columna que desee cambiar y pulse **Cambiar**. Se abrirá la ventana Cambiar columna.
4. Seleccione el tipo de datos.
5. Pulse en **Aceptar** para cambiar el tipo de datos y cierre la ventana.
6. Pulse **Aceptar** para modificar el apodo y cierre el cuaderno.

Para tratar el contenido de una columna local que tiene datos de tipo carácter en forma de datos bit (binarios), utilice la cláusula FOR BIT DATA de la sentencia ALTER NICKNAME. Cuando se utiliza esta cláusula para modificar el tipo de datos local de una columna, las conversiones de páginas de códigos no se realizan cuando se intercambia datos con otros sistemas. Las comparaciones se realizan con datos binarios, independientemente de la secuencia de clasificación de la base de datos remota.

Modificación de un tipo local para un objeto de origen de datos - ejemplos

Este tema proporciona ejemplos que muestran cómo modificar los tipos de datos para un objeto de origen de datos.

Ejemplo: Una correlación de tipo de datos numérico

En una tabla de Oracle de información sobre empleados, la columna BONUS se define con un tipo de datos NUMBER (32,3). El tipo de datos NUMBER (32,3) de Oracle se correlaciona por omisión con el tipo de datos DOUBLE de DB2, un tipo de datos de números de coma flotante de doble precisión. Una consulta que incluya la columna BONUS puede devolver valores parecidos a los siguientes:

```
5.00000000000000E+002  
1.00000000000000E+003
```

La notación científica indica el número de posiciones decimales y la dirección hacia la que debería desplazarse la coma decimal. En este ejemplo, +002 representa que la coma decimal debería desplazarse dos posiciones hacia la derecha y +003 significa que la coma decimal debería desplazarse tres posiciones hacia la derecha.

Las consultas que incluyen la columna BONUS pueden devolver valores parecidos a cantidades en dólares. Debe cambiar la definición local para la columna BONUS en la tabla del tipo de datos DOUBLE al tipo de datos DECIMAL. Utilice la precisión y la escala que reflejen el formato real de los bonos. Por ejemplo, si la porción de dólar de los bonos no debe sobrepasar las cinco cifras, correlacione

NUMBER (32,3) con DECIMAL (8,2). Bajo la restricción de este nuevo tipo local, las consultas que incluyan la columna BONUS devolverán valores como este:

```
500,00  
1000,00
```

El apodo para la tabla de Oracle es ORASALES. Para correlacionar la columna BONUS de la tabla ORASALES con el tipo de datos DECIMAL (8,2) de DB2, emita la siguiente sentencia ALTER NICKNAME:

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

ORASALES

El apodo que se ha definido para la table de Oracle.

ALTER COLUMN BONUS

El nombre de la columna que se ha definido de forma local en la vista de catálogo SYSCAT.COLUMNS de la base de datos federada.

LOCAL TYPE DECIMAL(8,2)

Identifica el nuevo tipo local para la columna.

Esta correlación se aplica solamente a la columna BONUS de la tabla de Oracle que se identifica con el apodo ORASALES. El resto de objetos de origen de que incluyan la columna BONUS utilizan la correlación de tipos de datos por omisión para el tipo de datos NUMBER de Oracle.

Ejemplo: Una correlación de tipos de datos

El apodo para la tabla de Oracle denominada SALES es ORASALES. La tabla SALES contiene una columna que es el tipo de datos DATE de Oracle. La correlación de tipos por omisión para el tipo de datos DATE de Oracle será con el tipo de datos TIMESTAMP de DB2. No obstante, le interesa visualizar solamente el valor de los datos al recuperar los datos de esta columna. Puede modificar el apodo de la tabla SALES para cambiar el tipo local por el tipo de datos DATE de DB2.

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

Ejemplo: Una correlación de tipos de datos para un origen de datos no relacional

El apodo para un archivo con estructura de tabla denominado drugdata1.txt es DRUGDATA1. El archivo drugdata1.txt contiene una columna que lista los nombres de fármacos. El nombre de la columna es DRUG. La columna DRUG se definió originalmente como CHAR(20). La longitud de la columna debe cambiarse a CHAR(30). Puede modificar el apodo para el archivo drugdata1.txt para cambiar la correlación a la longitud correcta:

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

Modificación de un tipo local para un objeto de origen de datos (línea de mandatos de DB2)

Para modificar un tipo de datos local debe utilizar la sentencia ALTER NICKNAME en lugar de la sentencia CREATE TYPE MAPPING. Puede modificar el tipo de datos desde el Centro de control de DB2 o desde la línea de mandatos de DB2.

Antes de empezar

El ID de autorización que emite la sentencia debe incluir como mínimo uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Privilegio ALTER en el apodo especificado en la sentencia
- Privilegio CONTROL en el apodo especificado en la sentencia
- Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
- Definidor del apodo tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el apodo.

Restricciones

Consulte el apartado “Restricciones en la modificación de apodos” en la página 34.

Acerca de esta tarea

Cuando se crea un apodo, los tipos de datos que están asociados al objeto de origen de datos se almacenan en la base de datos federada. Para algunos orígenes de datos, el derivador especifica los tipos de datos para el usuario. Para otros orígenes de datos, deberá especificar los tipos de datos cuando cree el apodo.

Puede especificar un tipo local para una columna de un objeto de origen de datos específico.

Importante: cambiar el tipo de datos local puede resultar en errores o pérdida de información si se modifica el tipo de datos local para una columna por un tipo que difiera significativamente de su tipo remoto.

Procedimiento

Para modificar el tipo de datos local desde el indicador de línea de mandatos, utilice la sentencia ALTER NICKNAME. Por ejemplo:

```
ALTER NICKNAME nickname ALTER COLUMN column_name  
LOCAL TYPE data_type
```

Para tratar el contenido de una columna local que tiene datos de tipo carácter en forma de datos bit (binarios), utilice la cláusula FOR BIT DATA de la sentencia ALTER NICKNAME. Cuando se utiliza esta cláusula para modificar el tipo de datos local de una columna, las conversiones de páginas de códigos no se realizan cuando se intercambia datos con otros sistemas. Las comparaciones se realizan con datos binarios, independientemente de la secuencia de clasificación de la base de datos remota.

Modificación de tipos de datos LONG por tipos de datos VARCHAR

Para habilitar operaciones de inserción y actualización para tipos de datos LONG, debe modificar los tipos de datos LONG por tipos de datos VARCHAR.

La Tabla 5 en la página 59 lista los tipos de datos LONG por origen de datos que pueden modificarse.

Tabla 5. Tipos de datos LONG por origen de datos que pueden modificarse por tipos de datos VARCHAR

Origen de datos	Tipo de datos remoto	Longitud	Tipo de datos local por omisión	ALTER to VARCHAR
DRDA	LONG VARCHAR	1-32672	CLOB	VARCHAR
	LONG VARCHAR FOR BIT DATA	1-32672	BLOB	VARCHAR FOR BIT DATA
Informix	BYTE	1-32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1-32672	CLOB	VARCHAR
Microsoft SQL Server	IMAGE	1-32672 host vars; 1-8000 literal	BLOB	VARCHAR FOR BIT DATA
	TEXT	1-32672 host vars; 1-8000 literal	CLOB	VARCHAR
Oracle NET8	LONG	1-32672 host vars; 1-4000 literal	CLOB	VARCHAR
	LONG RAW	1-32672 host vars; 1-4000 literal	BLOB	VARCHAR FOR BIT DATA
Sybase CTLIB	IMAGE	1-32672	BLOB	VARCHAR FOR BIT DATA
	TEXT	1-32672	CLOB	VARCHAR
Teradata	BYTE	32673-64000	BLOB	VARCHAR FOR BIT DATA(32672)
	CHAR	32673-64000	CLOB	VARCHAR(32672)
	VARBYTE	32673-64000	BLOB	VARCHAR FOR BIT DATA(32672)
	VARCHAR	32673-64000	CLOB	VARCHAR(32672)

Capítulo 4. Correlaciones de funciones y funciones definidas por el usuario

Las correlaciones de funciones asocian funciones de servidor federado y funciones definidas por el usuario con las funciones existentes en el origen de datos.

Correlaciones de funciones en un sistema federado

IBM InfoSphere Federation Server proporciona correlaciones predeterminadas entre funciones de origen de datos existentes y funciones equivalentes de DB2.

Para que el servidor federado utilice una función de origen de datos, se necesita una correlación desde una función de DB2 o una plantilla de función para la función de origen de datos.

Las correlaciones de funciones predeterminadas están en los módulos de derivador.

Para orígenes de datos no relacionales, no se pueden alterar temporalmente las correlaciones de funciones existentes ni crear nuevas correlaciones.

Funcionamiento de las correlaciones de funciones en un sistema federado

Cuando se envían consultas al servidor federado que contienen una o más funciones, el servidor federado busca información sobre las correlaciones entre las funciones de DB2 y las funciones de origen de datos.

El servidor federado comprueba dos lugares en busca de información de correlación:

- El derivador. El derivador de origen de datos contiene las correlaciones de funciones predeterminadas.
- La vista de catálogo SYSCAT.FUNCMAPPINGS. Esta vista contiene entradas que se crean que alteran temporalmente o aumentan las correlaciones de funciones predeterminadas que están en el derivador. También contiene nuevas correlaciones que se crean cuando no hay ninguna correlación de funciones. Cuando múltiples correlaciones se pueden aplicar a una función, se aplica la correlación creada más recientemente.

Las opciones de correlación de funciones especifican información sobre la función y sobre el coste potencial de procesar una función en el origen de datos. Las opciones de correlación de funciones proporcionan información como por ejemplo:

- Nombre de la función de origen de datos remoto
- El número estimado de instrucciones procesadas la primera y última vez que se ha invocado la función de origen de datos
- Número estimado de entradas y salidas que se realizan la primera y la última vez que se invoca la función de origen de datos.
- Número estimado de instrucciones que se procesan por cada invocación de la función de origen de datos.

Cuando crea una correlación de funciones, se realiza la correlación desde una función o plantilla de función de DB2 a una función equivalente en el origen de

datos. Si no existe una función equivalente de DB2 o si desea forzar que el servidor federado utilice la función de origen de datos, puede crear una plantilla de función que actúe como el equivalente.

¿Por qué son importantes las correlaciones de funciones?

Las correlaciones de funciones son una de las entradas importantes para el análisis de envío realizado por el optimizador de consultas.

Al decidir el mejor plan de acceso a consultas, el optimizador de consultas factoriza la capacidad del origen de datos para realizar un tipo concreto de función u operación de SQL. Si la función no tiene correlación, no se enviará al origen de datos para el proceso. Las funciones y otras operaciones que se pueden enviar al origen de datos mejorarán el rendimiento.

Si el origen de datos tiene una función similar a una función de DB2, pero devuelve resultados ligeramente diferentes, la creación de una correlación de funciones podría mejorar el rendimiento. Por ejemplo, la función STDEV (desviación estándar) de Informix produce resultados diferentes de los de la función STDDEV de DB2 para algunos conjuntos de datos de entrada. Por este motivo, el derivador Informix no tiene ninguna correlación predeterminada entre estas dos funciones. Si las diferencias del conjunto de resultados no tienen importancia, podría mejorar el rendimiento de consultas que acceden a los orígenes de datos de Informix y utilizar la función STDDEV de DB2. Al crear una correlación de funciones entre la función STDEV de Informix y la función STDDEV de DB2, se proporciona al optimizador de consultas la posibilidad de enviar el proceso de dicha función al origen de datos.

Cuándo crear correlaciones de funciones

Cuando la correlación de funciones predeterminadas no esté disponible para una función de origen de datos, puede crear una correlación de funciones.

Puede que la correlación de funciones no esté disponible porque la base de datos federada no tenga ninguna función que corresponda a la función de origen de datos.

Otro motivo por el cual una correlación puede no estar disponible es que el origen de datos tiene una función similar a una función de DB2, pero no devuelve los mismos resultados. Si el origen de datos devuelve resultados ligeramente diferentes o resultados diferentes para ciertos conjuntos de datos de entrada, los derivadores no correlacionarán con dichas funciones. Sin embargo, si las diferencias en los conjuntos de resultado no son importantes, se pueden crear correlaciones entre las funciones. La creación de una correlación puede mejorar el rendimiento.

Utilice correlaciones de funciones cuando:

- Una nueva función incorporada está disponible en el origen de datos
- Una nueva función definida por el usuario está disponible en el origen de datos
- No existe ninguna función equivalente de DB2
- Existe una función equivalente pero devuelve resultados ligeramente diferentes que no son de interés del usuario.

Los valores para correlaciones de funciones están almacenados en la vista de catálogo SYSCAT.FUNCMAPPINGS.

Cuando se cree una correlación de funciones, es posible que los valores devueltos desde una función evaluada en la fuente de datos sean diferentes de los valores devueltos desde una función compatible evaluada en la base de datos federada. IBM InfoSphere Federation Server utiliza la correlación de funciones, pero puede provocar un error de sintaxis de SQL o devolver resultados inesperados.

Funciones definidas por el usuario en aplicaciones

Los desarrolladores de aplicaciones a menudo necesitan crear sus propios conjuntos de funciones específicas para su aplicación o dominio. Pueden utilizar funciones escalares definidas por el usuario con este fin.

Por ejemplo, un almacén al por menor puede definir un tipo de datos PRICE para hacer un seguimiento del coste de los artículos que vende. Este almacén también puede querer definir una función SALES_TAX. Esta función tomaría el valor de un precio determinado como entrada, calcularía los impuestos aplicables y devolvería estos datos al usuario o aplicación solicitante.

Estas funciones pueden funcionar con todos los tipos de base de datos, incluyendo los tipos de objetos grandes y los tipos diferenciados. Las funciones definidas por el usuario permiten a las consultas contener potentes predicados de computación y búsqueda para filtrar datos irrelevantes cercanos a la fuente de los datos, reduciendo por tanto el tiempo de respuesta. El optimizador de SQL trata las funciones definidas por el usuario exactamente como funciones incorporadas tales como SUBSTR y LENGTH. Puede desarrollar aplicaciones utilizando distintos entornos de lenguaje de aplicación como, por ejemplo, C, C++ y COBOL. Las aplicaciones pueden compartir un conjunto de funciones definidas por el usuario incluso aunque se desarrollen utilizando entornos de lenguaje de aplicación distintos.

Las funciones definidas por el usuario pueden manipular datos y realizar acciones. Por ejemplo, puede habilitar una función definida por el usuario para enviar un mensaje electrónico o para actualizar un archivo plano.

En DB2, las funciones definidas por el usuario pueden incluir:

- Funciones que defina de cero.
- Funciones en el esquema SYSFUN. Los ejemplos incluyen funciones matemáticas tales como SIN, COS y TAN; funciones científicas como por ejemplo RADIANS, LOG10 y POWER; y funciones de propósito general como por ejemplo LEFT, DIFFERENCE y UCASE.

Requisitos para correlacionar funciones definidas por el usuario

Antes de poder invocar una función definida por el usuario de origen de datos en un sistema federado, la base de datos federada debe asociar la función de origen de datos con una especificación de función almacenada en el catálogo global del servidor federado.

Existen dos condiciones bajo las cuales la base de datos federada puede asociar una especificación de función con una función de origen de datos:

- La base de datos federada tiene una función cuya *signatura* corresponde a la *signatura* de la función de origen de datos. Una *signatura* consta de un nombre de función y de parámetros de entrada de función. Las *signaturas* se *corresponden* cuando las dos condiciones siguientes son ciertas:

- Contienen los mismos nombres y el mismo número de parámetros
- El tipo de datos de cada parámetro en una signatura es el mismo que (o se puede convertir al mismo) el tipo de datos del parámetro correspondiente en la otra signatura.
- Si la base de datos federada no tiene una función con la signatura necesaria, puede definir una plantilla de función que contenga esta signatura. A continuación, puede correlacionar la plantilla de función con la función de origen de datos que desee invocar.

Los valores para correlaciones de funciones están almacenados en la vista de catálogo SYSCAT.FUNCMAPPINGS.

Crear correlaciones de funciones

Utilice la sentencia CREATE FUNCTION MAPPING para especificar correlaciones de funciones alternativas que alteren temporalmente las correlaciones de funciones predeterminadas.

Cuando se crean correlaciones de funciones alternativas, las entradas aparecen en la vista de catálogo SYSCAT.FUNCMAPPINGS.

También puede utilizar la sentencia CREATE FUNCTION MAPPING para especificar opciones de correlación de funciones. Cuando especifica opciones de correlación de funciones, la información aparece en la vista de catálogo SYSCAT.FUNCMAPOPTIONS.

Con la sentencia CREATE FUNCTION MAPPING, puede:

- Crear una correlación de funciones para todos los orígenes de datos de un tipo específico. Por ejemplo, todos los orígenes de datos Informix.
- Crear una correlación de funciones para todos los orígenes de datos de un tipo y versión específicos. Por ejemplo, todos los orígenes de datos Informix 9.
- Crear una correlación de funciones para un servidor específico.
- Proporcionar información de estadísticas de correlación de funciones al optimizador
- Inhabilitar una correlación de funciones predeterminada o una correlación de funciones que haya definido.

Puede emitir la sentencia CREATE FUNCTION MAPPING en el procesador de línea de mandatos. También puede incorporar la sentencia CREATE FUNCTION MAPPING en un programa de aplicaciones. El Centro de control de DB2 no da soporte a la creación o modificación de correlaciones de funciones.

Especificación de nombres de función en la sentencia CREATE FUNCTION MAPPING

Los valores que se entren en la sentencia CREATE FUNCTION MAPPING dependen de si las funciones que esté correlacionando juntas tienen el mismo nombre o nombres diferentes.

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Correlación de funciones con el mismo nombre

Puede crear una correlación entre dos funciones (o una plantilla de función de DB2 y una función de origen de datos) que tienen el mismo nombre.

Procedimiento

Para correlacionar dos funciones con el mismo nombre, emita la sentencia CREATE FUNCTION MAPPING.

Ejemplo: Desea correlacionar una función definida por el usuario llamada MYFUN en un origen de datos de Informix con la función definida por el usuario de DB2 llamada TINA.MYFUN. El servidor del origen de datos de Informix se llama INFORMIX2. La siguiente sentencia correlaciona la función:

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

Correlación de funciones con nombres diferentes

Puede crear una correlación entre dos funciones (o una plantilla de función de DB2 y una función de origen de datos) que tienen nombres diferentes.

Para crear una correlación entre dos funciones con nombres diferentes, emita la sentencia CREATE FUNCTION MAPPING:

1. Asigne el nombre de la función DB2 o de la plantilla de función al parámetro `function_name`.
2. Especifique una opción de correlación de funciones denominada `REMOTE_NAME` y asigne el nombre de la función de origen de datos a esta opción. `REMOTE_NAME` no debe tener más de 255 caracteres.

Ejemplo: Desea correlacionar una función definida por el usuario denominada UPPERCASE en un origen de datos de Oracle con la función de DB2 UCASE(CHAR). El servidor del origen de datos de Oracle se denomina ORACLE2. Decide llamar a esta correlación de funciones ORACLE_UPPER. La sintaxis sería:

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

Creación de una correlación de funciones para un tipo de origen de datos específico

Se puede crear una correlación con una función para todos los orígenes de datos de un tipo específico.

Antes de empezar

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Restricciones

No puede sobrescribir las correlaciones de funciones existentes ni crear correlaciones nuevas para orígenes de datos no relacionales.

Procedimiento

Para correlacionar una plantilla de función de DB2 con una función de origen de datos, utilice la sentencia CREATE FUNCTION MAPPING.

Ejemplo: Correlacionar una plantilla de función de DB2 con una función definida por el usuario de Oracle para todos los orígenes de datos de Oracle

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

La plantilla se denomina STATS y pertenece a un esquema denominado NOVA. La función definida por el usuario de Oracle se denomina STATISTICS y pertenece a un esquema denominado STAR.

Creación de una correlación de funciones para un tipo de origen de datos y una versión específicos

Se puede crear una correlación con una función para todos los orígenes de datos que utilicen una versión específica del tipo de origen de datos.

Antes de empezar

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Restricciones

No puede sobrescribir las correlaciones de funciones existentes ni crear correlaciones nuevas para orígenes de datos no relacionales.

Procedimiento

Para crear una correlación para un tipo y una versión de origen de datos específica, utilice la sentencia CREATE FUNCTION MAPPING.

Ejemplo: Correlacionar una plantilla de función de DB2 con una función definida por el usuario de Sybase para todos los orígenes de datos de Sybase que utilizan la Versión 15

La plantilla se denomina SYB_STATS y pertenece a un esquema denominado EARTH. La función definida por el usuario de Sybase se denomina STATISTICS y pertenece a un esquema denominado MOON. CREATE FUNCTION MAPPING es:

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

Creación de una correlación de funciones para todos los objetos de origen de datos en un servidor específico

Se puede crear una correlación con una función para todos los orígenes de datos de un tipo específico.

Antes de empezar

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Restricciones

No puede sobrescribir las correlaciones de funciones existentes ni crear correlaciones nuevas para orígenes de datos no relacionales.

Procedimiento

Para crear una correlación de funciones para todos los objetos de origen de datos en un servidor específico

Ejemplo Correlacionar una plantilla de función denominada BONUS con una función definida por el usuario denominada BONUS

Sólo se quiere que la correlación se aplique al servidor de orígenes de datos de Oracle denominado ORA_SALES. Puesto que los nombre de función son iguales, no es necesario especificar la opción de correlación de funciones REMOTE_NAM.

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

Plantillas de función

El servidor federado reconoce una función de origen de datos cuando existe una correlación entre la función de origen de datos y una función equivalente de DB2 en la base de datos federada.

Puede crear una plantilla de función para que actúe como función equivalente de DB2 cuando no exista un equivalente.

Una *plantilla de función* es una función de DB2 que se crea con el fin de forzar al servidor federado para que invoque una función de origen de datos. Sin embargo, a diferencia de una función normal, una plantilla de funciones no tiene código ejecutable. Cuando el servidor federado recibe consultas que especifican la plantilla de funciones, el servidor federado invocará la función de origen de datos.

La plantilla de funciones se crea con la sentencia CREATE FUNCTION utilizando el parámetro AS TEMPLATE.

Después de crear una plantilla de funciones, debe crear la correlación de funciones entre la plantilla y la función de origen de datos. Una correlación de funciones se crea utilizando la sentencia CREATE FUNCTION MAPPING.

Creación de plantillas de funciones

El servidor federado reconoce una función de origen de datos cuando hay una correlación entre la función de origen de datos y una función equivalente en la base de datos federada. Puede crear una plantilla de función para que actúe de función equivalente cuando ésta no exista.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autorización SYSADM o DBADM
- Autorización IMPICIT_SCHEMA en las bases de datos, si el nombre de esquema implícito o explícito de la función no existe
- Privilegio CREATEIN en el esquema, si existe el nombre de esquema de la función.

Restricciones

Si la función de origen de datos tiene valores de entrada:

- La función equivalente de DB2 debe tener el mismo número de parámetros de entrada que la función de origen de datos.
- Los tipos de datos de los parámetros de entrada para la función equivalente de DB2 deben ser compatibles con los tipos de datos correspondientes de los parámetros de entrada para la función de origen de datos. Los tipos de datos no pueden ser LONG VARCHAR, LONG VARGRAPHIC o un tipo definido por el usuario.

Si la función de origen de datos no tiene parámetros de entrada, la función equivalente de DB2 no puede tener ningún parámetro de entrada.

Procedimiento

Para crear una plantilla de función:

1. Utilice la sentencia CREATE FUNCTION con el parámetro AS TEMPLATE. Por ejemplo:

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

BONUS ()

El nombre que se dio a la plantilla de función.

RETURNS *DECIMAL(8,2)*

El tipo de datos de salida.

AS TEMPLATE

Indica que es una plantilla de función y no una función.

DETERMINISTIC

Especifica que la función siempre devuelve los mismos resultados para un conjunto de valores de argumento proporcionado.

NO EXTERNAL ACTION

Especifica que la función no tiene impacto externo en los objetos que no están gestionados por el gestor de base de datos.

Debe especificar las cláusulas DETERMINISTIC i NO EXTERNAL ACTION dependiendo de si la función es determinante y si provoca alguna acción externa. En caso contrario, las restricciones se impondrán en las operaciones SQL a las que se da soporte con esta plantilla de función.

2. Después de crear una plantilla de funciones, debe crear la correlación de funciones entre la plantilla y la función de origen de datos. Una correlación de funciones se crea utilizando la sentencia CREATE FUNCTION MAPPING. Por ejemplo:

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

MY_INFORMIX_FUN

El nombre que se dio a la correlación de funciones. El nombre no puede duplicar un nombre de correlación de funciones que ya esté descrito en el catálogo global de bases de datos federadas. Debe ser exclusivo.

FOR *BONUS()*

El nombre de la plantilla de función de DB2 local. Incluye valores de entrada de tipos de datos entre paréntesis.

SERVER TYPE *INFORMIX*

Identifica el tipo de origen de datos que contiene la función a la que desea correlacionar.

OPTIONS (*REMOTE_NAME 'BONUS()'*)

Una opción que identifica el nombre de la función de origen de datos remoto que está correlacionando con la plantilla de función de DB2 local.

Inhabilitación de una correlación de funciones predeterminada

La correlación de funciones predeterminada no puede descartarse. Sin embargo, puede hacer que sean inoperativos inhabilitándolos.

Antes de empezar

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Procedimiento

Para inhabilitar una correlación de funciones por omisión, la sentencia CREATE FUNCTION MAPPING específica el nombre de la función de DB2 y establece la opción DISABLE en 'Y'.

Ejemplo: Inhabilitar una correlación de funciones por omisión entre la función SIN de DB2 SIN y una función similar en orígenes de datos de Oracle

Cuando se procesa una consulta que necesita datos de Oracle y que hace referencia a SIN, puede que se invoquen ambas funciones. La función invocada depende de qué función considere el optimizador de consultas que necesita menos actividad general

Para asegurarse de que la función SIN de DB2 se ha invocado y de que la función SIN de Oracle no se ha invocado, debe inhabilitar la correlación de funciones por omisión. Utilice la sintaxis siguiente:

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

Descarte de una correlación de funciones

Cuando ya no se necesite una correlación de funciones que haya creado, puede suprimirla ejecutando la sentencia DROP.

Antes de empezar

Los privilegios contenidos en el ID de autorización de la sentencia deben tener autoridad SYSADM o DBADM.

Acerca de esta tarea

Si se descarta una correlación de funciones que ha creado para alterar temporalmente una correlación de funciones predeterminada, se utilizará la correlación de funciones predeterminada.

Las correlaciones de funciones se listan en la vista de catálogo SYSCAT.FUNCMAPPINGS.

Procedimiento

Para descartar una correlación de funciones que haya creado, utilice la sentencia DROP:

Ejemplo: Descartar una correlación de funciones denominada BONUS_CALC
DROP FUNCTION MAPPING *BONUS_CALC*

Capítulo 5. Creación de especificaciones de índice

Para orígenes de datos relacionales, cree especificaciones de índice para almacenar información acerca de las columnas de índices remotos en el catálogo global y para asegurar un rendimiento óptimo de las consultas de búsqueda.

Especificaciones de índice en un sistema federado

En un sistema federado, se utiliza la sentencia `CREATE INDEX` con un apodo para almacenar información en el catálogo global sobre la disponibilidad de un índice en el objeto remoto. El optimizador de consultas utiliza esta información para optimizar consultas.

Cuando se emite una sentencia `CREATE INDEX`

- Si se crea un apodo para una tabla, la sentencia `CREATE INDEX` recopila información de índice sobre el índice que se ha creado en el tabla remota.
- Si se crea un apodo para una vista, la sentencia `CREATE INDEX` hace referencia al apodo para la vista y contiene información sobre el índice en la tabla subyacente a la vista.

La especificación de índice notifica al servidor federado sobre las columnas y sus propiedades de exclusividad que comprenden un índice remoto. No notifica al servidor federado sobre las propiedades estadísticas del índice como, por ejemplo, el número de valores exclusivos de la clave de índice.

No necesita suministrar especificaciones de índice si el índice remoto estaba en su lugar en el momento en el que el índice se ha creado.

Un servidor federado no crea una especificación de índice cuando el usuario crea un apodo para:

- Una tabla que no tiene ningún índice
- Una vista, que normalmente no tiene información de índice almacenada en el catálogo remoto
- Un objeto de origen de datos que no tiene un catálogo remoto del que el servidor federado pueda obtener información de índice

Imaginemos que una tabla adquiere un nuevo índice, además de los que ya tenía al crearse el apodo. Puesto que la información de índice se proporciona al catálogo global durante la creación del apodo, el servidor federado no reconoce el nuevo índice. De forma similar, cuando se crea un apodo para una vista, el servidor federado no reconoce la tabla subyacente (ni los índices de ésta) a partir de la cual se ha generado la vista. En estas circunstancias, puede proporcionar la información de índice necesaria al catálogo global. Puede crear una especificación de índice para las tablas que no tienen ningún índice. La especificación de índice indica al optimizador de consultas en qué columna o columnas de la tabla debe buscar para encontrar los datos rápidamente.

Utilice especificaciones de índice con orígenes de datos relacionales. La creación de una especificación de índice para un origen de datos no relacional no mejorará el rendimiento.

Creación de especificaciones de índice para objetos de origen de datos

Al crearse un apodo para la tabla del origen de datos, el servidor federado suministra al catálogo global la información sobre los índices de la tabla de origen de datos. El optimizador utiliza la información para agilizar el proceso de solicitudes distribuidas. Dicha información consiste en un conjunto de metadatos y se denomina *especificación de índice*.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autorización SYSADM o DBADM
- Un privilegio CONTROL en el objeto o un privilegio INDEX en el objeto. Y uno de autorización IMPLICIT_SCHEMA en la base de datos, si el nombre implícito o explícito de nombre de esquema del índice no existe, o el privilegio CREATEIN en el esquema, si el nombre de esquema del índice hace referencia a un esquema existente.

Restricciones

Hay ciertas restricciones a la hora de crear un índice en un apodo.

- Si se aplica la opción de vinculación DYNAMICRULES BIND, la sentencia no puede prepararse dinámicamente. Asimismo, no puede utilizar los parámetros INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS y ALLOW REVERSE SCANS en la sentencia CREATE INDEX.
- Sólo debe especificarse UNIQUE si los datos para la clave de índice contienen valores exclusivos para todas las filas de la tabla de origen de datos. La exclusividad no se someterá a comprobación.
- La suma de las longitudes almacenadas de las columnas especificadas no debe superar 1024.
- No puede utilizarse como parte de un índice ninguna columna LOB ni ninguna columna de tipo diferenciado basado en LOB. Dicha restricción se impone incluso si el atributo de longitud de la columna es suficientemente pequeño para ajustarse al límite de 1024.

Acerca de esta tarea

El servidor federado no creará ninguna especificación de índice si:

- Se crea un apodo para una tabla sin índice.
- Se crea un apodo para un objeto de origen de datos que no contenga índices tales como una vista o un sinónimo de Informix.
- Se crea un apodo para un objeto no relacional, por ejemplo, un archivo de tabla estructurada, una hoja de cálculo Excel, o un archivo con identificador XML.
- El índice remoto está en una columna LOB o una columna XML.
- El índice remoto tiene una longitud de clave total superior a 1024 bytes.
- El número máximo de partes clave es superior a 16.

En estas circunstancias el servidor federado no almacena especificaciones de índice para los objetos de origen de datos. Sin embargo, para los dos primeros elementos

de la lista anterior se puede proporcionar la información de índice al catálogo global. Puede utilizar la sentencia CREATE INDEX para especificar la información de índice.

Procedimiento

Para crear un índice, incruste la sentencia CREATE INDEX en un programa de aplicación o emita la sentencia como sentencia dinámica de SQL desde el Centro de control o la línea de mandatos.

Al utilizar mandatos, la sentencia CREATE INDEX crea una especificación de índice en el catálogo global federado; no crea ningún índice en la tabla de origen de datos.

Utilice la sintaxis siguiente para crear una especificación de índice:

```
CREATE INDEX index_name ON nickname
(column_name) SPECIFICATION ONLY
CREATE UNIQUE INDEX index_name ON nickname
(column_name DESC) SPECIFICATION ONLY
```

Para una especificación de índice, *column_name* es el nombre con que el servidor federado hace referencia a una columna de la tabla de origen de datos.

Creación de especificaciones de índice sobre tablas que adquieren índices nuevos

Para situaciones en las que una tabla adquiere un índice nuevo, se debería crear una especificación de índice en el apodo que corresponde a la tabla.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autorización SYSADM o DBADM
- Un privilegio CONTROL en el objeto o un privilegio INDEX en el objeto. Y uno de autorización IMPLICIT_SCHEMA en la base de datos, si el nombre implícito o explícito de nombre de esquema del índice no existe, o el privilegio CREATEIN en el esquema, si el nombre de esquema del índice hace referencia a un esquema existente.

Restricciones

Hay ciertas restricciones a la hora de crear un índice en un apodo.

- Si se aplica la opción de vinculación DYNAMICRULES BIND, la sentencia no puede prepararse dinámicamente. Asimismo, no puede utilizar los parámetros INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS y ALLOW REVERSE SCANS en la sentencia CREATE INDEX.
- Sólo debe especificarse UNIQUE si los datos para la clave de índice contienen valores exclusivos para todas las filas de la tabla de origen de datos. La exclusividad no se someterá a comprobación.
- La suma de las longitudes almacenadas de las columnas especificadas no debe superar 1024.

- No puede utilizarse como parte de un índice ninguna columna LOB ni ninguna columna de tipo diferenciado basado en LOB. Dicha restricción se impone incluso si el atributo de longitud de la columna es suficientemente pequeño para ajustarse al límite de 1024.

Acerca de esta tarea

Existen ciertas situaciones en que una tabla adquiere un índice nuevo:

- Se crea un apodo para una tabla que no tiene índice, pero adquiere un índice después.
- Se crea un apodo para una tabla que no tiene índice, pero adquiere un índice después.

En estas situaciones, se debería crear una especificación de índice para la tabla de modo que SQL Compiler pueda utilizar dicha información al procesar las consultas que hacen referencia a la tabla.

Procedimiento

Los ejemplos siguientes describen cómo crear una especificación de índice en un apodo que se corresponda a una tabla que adquiere un índice.

Ejemplo: Una tabla no tiene índice, lo adquiere después.

Supongamos que se crea el apodo *EMPLOYEE* para una tabla de origen de datos denominada *CURRENT_EMP*, que no tiene índices. A veces cuando se crea el apodo, se define un índice en *CURRENT_EMP* mediante las columnas *WORKDEPT* y *JOB* para la clave de índice.

Para crear una especificación de índice que describa el índice, la sintaxis sería:

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

en que *JOB_BY_DEPT* es el nombre del índice.

Ejemplo: Una tabla adquiere un índice nuevo

Supongamos que se crea el apodo *JP_SALES* para una tabla denominada *JAPAN_SALES*. Un índice nuevo se añade después a la tabla sumándose a los que tenía desde el momento en que se creó el apodo. El índice nuevo utiliza la columna *MARKUP* para la clave de índice.

Para crear una especificación de índice que describa el índice, la sintaxis sería:

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

en que *JP_MARKUP* es el nombre de índice.

Creación de especificaciones de índice en vistas

Cuando se crea un apodo para una vista, el servidor federado no es consciente de la tabla subyacente, ni de los índices, desde los que se creó la vista. Se debería crear una especificación de índice para la vista de modo que SQL Compiler pueda utilizar dicha información al procesar las consultas que hacen referencia a la vista.

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autorización SYSADM o DBADM
- Un privilegio CONTROL en el objeto o un privilegio INDEX en el objeto. Y uno de autorización IMPLICIT_SCHEMA en la base de datos, si el nombre implícito o explícito de nombre de esquema del índice no existe, o el privilegio CREATEIN en el esquema, si el nombre de esquema del índice hace referencia a un esquema existente.

Restricciones

Hay ciertas restricciones a la hora de crear un índice en un apodo.

- Si se aplica la opción de vinculación DYNAMICRULES BIND, la sentencia no puede prepararse dinámicamente. Asimismo, no puede utilizar los parámetros INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS y ALLOW REVERSE SCANS en la sentencia CREATE INDEX.
- Sólo debe especificarse UNIQUE si los datos para la clave de índice contienen valores exclusivos para todas las filas de la tabla de origen de datos. La exclusividad no se someterá a comprobación.
- La suma de las longitudes almacenadas de las columnas especificadas no debe superar 1024.
- No puede utilizarse como parte de un índice ninguna columna LOB ni ninguna columna de tipo diferenciado basado en LOB. Dicha restricción se impone incluso si el atributo de longitud de la columna es suficientemente pequeño para ajustarse al límite de 1024.

Procedimiento

Para crear una especificación de índice para una vista:

- Asegúrese de que la columna o columnas en que se basa el índice de tabla, forme parte de la vista.
- Si se quieren crear especificaciones de índice para todos los índices en la tabla subyacente, cada especificación de índice debe crearse de modo independiente.

Ejemplo: Crear una especificación que describa el índice REGION

Supongamos que se crea el apodo *JP_SALES2007* para una tabla denominada JAPAN_SALES2007. La tabla subyacente para esta vista es la tabla JAPAN_SALES que contiene varios índices: REGION, AMOUNT, SALES_REP. La sentencia CREATE INDEX que se ha creado hará referencia al apodo para la vista y contendrá información sobre el índice de la tabla subyacente para la vista.

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

donde *JP_2007_REGION* es el nombre de índice y *JP_SALES2007* es el apodo para la vista JAPAN_SALES2007.

Creación de especificaciones de índice sobre sinónimos de Informix

El tema describe la acción que el servidor federado toma para Informix basado en una tabla o en una vista:

Antes de empezar

El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:

- Autorización SYSADM o DBADM
- Un privilegio CONTROL en el objeto o un privilegio INDEX en el objeto. Y uno de autorización IMPLICIT_SCHEMA en la base de datos, si el nombre implícito o explícito de nombre de esquema del índice no existe, o el privilegio CREATEIN en el esquema, si el nombre de esquema del índice hace referencia a un esquema existente.

Restricciones

Hay ciertas restricciones a la hora de crear un índice en un apodo.

- Si se aplica la opción de vinculación DYNAMICRULES BIND, la sentencia no puede prepararse dinámicamente. Asimismo, no puede utilizar los parámetros INCLUDE, CLUSTER, PCTFREE, MINPCTUSED, DISALLOW REVERSE SCANS y ALLOW REVERSE SCANS en la sentencia CREATE INDEX.
- Sólo debe especificarse UNIQUE si los datos para la clave de índice contienen valores exclusivos para todas las filas de la tabla de origen de datos. La exclusividad no se someterá a comprobación.
- La suma de las longitudes almacenadas de las columnas especificadas no debe superar 1024.
- No puede utilizarse como parte de un índice ninguna columna LOB ni ninguna columna de tipo diferenciado basado en LOB. Dicha restricción se impone incluso si el atributo de longitud de la columna es suficientemente pequeño para ajustarse al límite de 1024.

Acerca de esta tarea

En Informix, puede crear un sinónimo para una tabla o vista. El servidor federado le permite crear apodos para sinónimos de Informix, en cambio la acción que realiza el servidor federado depende de si el sinónimo se basa en una tabla o en una vista:

- Supongamos que un apodo se crea para un sinónimo y éste se basa en una tabla de Informix. Si el servidor federado determina que la tabla a la que hace referencia el sinónimo tiene un índice, entonces se creará una especificación de índice para el sinónimo. Si la tabla a la que hace referencia el sinónimo no tiene índice, no se creará ninguna especificación de índice para el sinónimo. Sin embargo, puede crear una especificación de índice de modo manual, mediante la sentencia CREATE INDEX.
- Supongamos que un apodo se crea para un sinónimo y éste se basa en una vista de Informix. El servidor federado no puede determinar en qué tabla o tablas subyacentes se basa la vista. Por consiguiente, no se creará ninguna especificación de índice para el sinónimo. Sin embargo, puede crear una especificación de índice de modo manual, mediante la sentencia CREATE INDEX.

Procedimiento

Los ejemplos siguientes describen cómo crear una especificación de índice en un apodo que se corresponda a un sinónimo de Informix.

Ejemplo: Se crea un apodo en un sinónimo de Informix basado en una tabla

Cuando el sinónimo se basa en una tabla de Informix que no contiene índice, puede crear una especificación de índice para que el sinónimo indique al optimizador en qué columna o columnas debe buscar para encontrar los datos rápidamente. La sentencia que se cree especificará el apodo para el sinónimo y se proporcionará la información sobre la columna o columnas de la tabla en que se basa el sinónimo.

En este ejemplo, se crea el apodo *CONTRACTS* para un sinónimo denominado *SALES_CONTRACTS*. La tabla en que se basa este sinónimo se llama *SALES2006_TABLE* y contiene varios índices: *REGION*, *AMOUNT*, *SALES_REP*. La sentencia *CREATE INDEX* que se ha creado hará referencia al apodo para el sinónimo y contendrá información sobre el índice de la tabla subyacente para el sinónimo.

Para crear una especificación de índice que describa el índice *REGION*, la sintaxis sería:

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

donde *NORTHWEST_2006_REGION* es el nombre de índice *CONTRACTS* es el apodo para el sinónimo *SALES_CONTRACTS*.

Ejemplo: Se crea un apodo en un sinónimo Informix basado en una vista

Se crea el apodo *JP_SALES2007* para un sinónimo basado en una vista denominada *JAPAN_SALES2007*. La tabla subyacente para esta vista es la tabla *JAPAN_SALES* que contiene varios índices: *REGION*, *AMOUNT*, *SALES_REP*. La sentencia *CREATE INDEX* que se ha creado hará referencia al apodo para el sinónimo y contendrá información sobre el índice de la tabla subyacente para la vista.

Cuando cree una especificación de índice para un sinónimo basado en una vista, asegúrese de que la columna o columnas en que se basa el índice de tabla, forme parte de la vista. Si se quieren crear especificaciones de índice para todos los índices en la tabla subyacente, cada especificación de índice debe crearse de modo independiente.

Para crear una especificación de índice que describa el índice *REGION*, la sintaxis sería:

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

donde *JP_2007_REGION* es el nombre de índice y *JP_SALES2007* es el apodo para la vista *JAPAN_SALES2007*.

Capítulo 6. Desarrollo de procedimientos federados

Los procedimientos federados permiten invocar procedimientos en un origen de datos como si el procedimiento remoto fuera un procedimiento local.

Procedimientos federados

Un *procedimiento federado* es un objeto de base de datos federada que hace referencia a un procedimiento de un origen de datos.

Los procedimientos federados no son nombres alternativos para procedimientos de origen como sí lo son los alias. Un procedimiento federado se define en la base de datos federada, pero llama a un procedimiento de origen de datos cuando se invoca el procedimiento federado. Dado que el procedimiento federado es un objeto de base de datos federada, los usuarios y las aplicaciones cliente pueden invocar la lógica del procedimiento de origen de datos llamando a un procedimiento federado. El procedimiento federado devuelve los resultados del procedimiento de origen de datos, como los parámetros de salida. Al utilizar un procedimiento federado, la ubicación del procedimiento de origen de datos es transparente para los usuarios y las aplicaciones cliente. Debe utilizar el nombre del procedimiento federado para llamar al procedimiento de origen.

Un procedimiento federado es para un procedimiento remoto lo que un apodo es para una tabla remota. Los apodos y los procedimientos federados son objetos de la base de datos federada. Un apodo es un objeto que hace referencia a un objeto, como una tabla o vista, del origen de datos. Con un apodo, puede realizarse una consulta de un objeto de origen de datos. Con un procedimiento federado, puede efectuarse una llamada a un procedimiento de origen de datos.

Para registrar un procedimiento federado, se utiliza la sentencia CREATE PROCEDURE (fuente), y para llamar a un procedimiento, se utiliza la sentencia CALL. Puede incluir la sentencia CREATE PROCEDURE (fuente) en un programa de aplicación o emitir la sentencia con sentencias SQL dinámicas.

Parámetros y tipos de datos

La federación utiliza tres tipos de parámetros: IN, OUT e INOUT. Al crear un procedimiento federado, se utilizan correlaciones de tipos de datos directas predeterminadas para correlacionar los tipos de datos para los parámetros del procedimiento de origen de datos con los tipos de datos federados. Esta correlación incluye tipos de datos para los parámetros de entrada y salida, y tipos de datos para las columnas del conjunto de resultados. Puede utilizar la sentencia CREATE TYPE MAPPING para sustituir la correlación de tipo predeterminada para los parámetros de origen. No obstante, las correlaciones de tipos del conjunto de resultados no se ven afectadas por las correlaciones de tipos definidos por el usuario. Puede utilizar todos los tipos de datos soportados para las columnas de apodo, excepto los tipos de datos LOB. Los procedimientos federados no dan soporte a los tipos de datos complejos.

Correlaciones y autorización de usuario

Para llamar a un procedimiento federado, debe tener las autorizaciones correctas en el procedimiento federado y el procedimiento de origen de datos. Al llamar a

un procedimiento federado, para acceder a las tablas de origen se utilizan la correlación y los privilegios de usuario del ID de autorización que ha creado el procedimiento federado.

Por ejemplo, el usuario ZELLER crea un procedimiento federado denominado FP1. El procedimiento FP1 hace referencia a un procedimiento Sybase que accede a una tabla Sybase. El ID de usuario remoto de la correlación de usuario para ZELLER tiene el privilegio de actualizar la tabla Sybase. El usuario ZELLER otorga el privilegio EXECUTE al usuario BHATIA en el procedimiento FP1. El usuario BHATIA debe tener una correlación de usuario válida a un ID de usuario remoto que tenga el privilegio EXECUTE en el procedimiento Sybase al que se hace referencia en el procedimiento FP1. El ID de usuario remoto con el que está correlacionado el usuario BHATIA no necesita tener el privilegio SELECT en el procedimiento Sybase. Cuando el usuario BHATIA llama al procedimiento FP1, el usuario BHATIA puede actualizar la tabla en Sybase.

Llamadas de procedimiento y niveles de acceso

Las llamadas de procedimiento y los niveles de acceso presentan estos problemas:

- De forma predeterminada, la cláusula CALL RESOLUTION IMMEDIATE se utiliza con procedimientos federados para emitir el mandato PRECOMPILE. La cláusula CALL RESOLUTION DEFERRED no está soportada en procedimientos federados.
- No es posible ver la salida cuando un procedimiento federado llama a un procedimiento de origen de datos que imprime los datos en un almacenamiento intermedio o en la salida estándar.
- Al llamar a un procedimiento federado desde una función externa definida por el usuario, el procedimiento federado no debe tener el nivel de acceso definido en READS SQL DATA o MODIFIES SQL DATA. El acceso federado está bloqueado dentro de las funciones externas.
- Las limitaciones que se aplican a las llamadas a los procedimientos locales en el modo de paso a través también se aplican a los procedimientos federados.
- Cada argumento de una sentencia CALL debe ser compatible con el parámetro correspondiente en el procedimiento. Los procedimientos federados siguen las mismas reglas de asignación de parámetros que los procedimientos locales.

Transacciones

Tenga en cuenta los siguientes puntos cuando utilice procedimientos federados con transacciones:

- El procedimiento de origen de datos al que hace referencia el procedimiento federado no debe emitir ninguna sentencia COMMIT ni ROLLBACK. La federación no aplica esta restricción y es posible que se produzcan incoherencias de datos si el procedimiento de origen de datos emite una sentencia COMMIT o ROLLBACK.
- Los procedimientos federados con el nivel de acceso MODIFIES SQL DATA no pueden invocarse dentro de desencadenadores, sentencias compuestas dinámicas, esalares SQL, tablas, funciones de fila ni métodos. Tras emitir una sentencia SAVEPOINT, no es posible llamar a un procedimiento federado con el nivel de acceso MODIFIES SQL DATA.
- Los procedimientos federados no dan soporte a cursores WITH HOLD, cursores escalares ni cursores actualizables. Si el procedimiento de origen de daos utiliza estos tipos de cursor, no recibirá ningún mensaje de aviso o error. Es posible que las aplicaciones que acceden a conjuntos de resultados con procedimientos de

origen de datos que utilizan estos tipos de cursor se comporten de forma distinta. Generalmente, los cursores se devuelven sin cursores de capacidad de retención y de sólo lectura directos.

Vistas de catálogo

Esta vistas de catálogo del sistema almacenan información sobre procedimientos almacenados:

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

Orígenes de datos y procedimientos federados

Antes de crear procedimientos federados, revise cómo la federación da soporte a los procedimientos de orígenes de datos.

Procedimientos federados para orígenes de datos DB2

Antes de crear un procedimiento federado, revise cómo especificar opciones en la sentencia CREATE PROCEDURE.

Al crear procedimientos federados, tenga en cuenta lo siguiente:

- Los procedimientos DB2 dan soporte a los parámetros IN, OUT e INOUT.
- Debe especificar las mismas opciones que el proceso de DB2 especifica para las cláusulas SQL de acceso, deterministas y de acción externa de la sentencia CREATE PROCEDURE.
- Si dos procedimientos DB2 tienen el mismo valor, utilice la opción NUMBER OF PARAMETERS para identificar el procedimiento que desea utilizar.
- La base de datos federada no emite ninguna sentencia COMMIT para los procedimientos federados que se crean para los procedimientos en DB2 Universal Database for z/OS y que contienen una cláusula COMMIT ON RETURN YES.

Ejemplo

Este ejemplo muestra cómo utilizar la sentencia CREATE PROCEDURE para crear un procedimiento federado para un procedimiento de origen de datos en 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

Necesario. Especifica el nombre del procedimiento federado.

SOURCE KELLER.PROC1_DB2

Necesario. Especifica el esquema y el nombre para el procedimiento DB2. Para procedimientos DB2, debe especificar un nombre de dos partes en la sentencia CREATE PROCEDURE. El formato de este nombre de dos partes es *nombre_esquema_origen.nombre_procedimiento_origen*.

NUMBER OF PARAMETERS 3

Especifica el número total de parámetros IN, OUT e INOUT que utiliza el

procedimiento DB2. Utilice este parámetro si tiene más de un procedimiento con el mismo nombre de esquema y nombre de procedimiento. Por ejemplo, si el esquema es KELLER y tiene un procedimiento PROC1 con tres parámetros y otro procedimiento PROC1 con un parámetro, el nombre para ambos procedimientos es KELLER.PROC1. El valor para NUMBER OF PARAMETERS en el procedimiento de origen de datos indica a qué procedimiento se hace referencia en la sentencia CREATE PROCEDURE.

FOR SERVER DB2_SERVER

Necesario. Especifica una definición de servidor en la que se crea el procedimiento federado.

SPECIFIC MYPROC1

Especifica un nombre exclusivo para el procedimiento federado que se está creando. Este parámetro sólo se utiliza para procedimientos federados y no está asociado con procedimientos de orígenes de datos. Si no especifica ningún nombre exclusivo, el gestor de bases de datos federado genera un nombre.

WITH RETURN TO CLIENT ALL

Especifica que el conjunto de resultados se devuelve a la aplicación cliente. La federación devuelve como máximo un conjunto de resultados. Si este parámetro no se especifica, el valor predeterminado es WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indica el nivel de acceso a datos para las sentencias SQL que se incluyen en el procedimiento federado. Si la cláusula especificada no coincide con el procedimiento DB2, se devuelve un mensaje de error. Si no especifica esta cláusula, se utiliza la cláusula para el procedimiento DB2.

DETERMINISTIC

Especifica si el procedimiento federado siempre devuelve los mismos resultados para un conjunto de valores de argumento proporcionado. Este parámetro puede mejorar el rendimiento de la interacción entre el servidor federado y el origen de datos. Si la cláusula especificada no coincide con el procedimiento DB2, se devuelve un mensaje de error. Si no especifica esta cláusula, se utiliza la cláusula para el procedimiento DB2.

EXTERNAL ACTION

Especifica si el procedimiento federado lleva a cabo una acción que cambia el estado de un objeto que no se gestiona a través del gestor de bases de datos. Si la cláusula especificada no coincide con el procedimiento DB2, se devuelve un mensaje de error. Si no especifica esta cláusula, se utiliza la cláusula para el procedimiento DB2.

Procedimientos federados y Microsoft SQL Server

Antes de crear un procedimiento federado, revise qué parámetros pueden utilizarse, cómo se devuelven los conjuntos de resultados y qué limitaciones existen.

Parámetros

Los procedimientos de Microsoft SQL Server dan soporte al uso de los parámetros opcionales INPUT y OUTPUT. El derivador SQL Server correlaciona cada parámetro INPUT con un parámetro IN federado, y cada parámetro OUTPUT, con un parámetro INOUT federado. Si utiliza los parámetros opcionales en un

procedimiento de SQL Server, debe contarlos cuando especifique el valor de la cláusula NUMBER OF PARAMETERS de la sentencia CREATE PROCEDURE.

Conjuntos de resultados

El derivador SQL Server puede devolver un conjunto de resultados y un parámetro OUTPUT. Cuando un procedimiento de SQL Server devuelve un parámetro OUTPUT y un conjunto de resultados, sólo se devuelve el parámetro. El conjunto de parámetros se descarta y se recibe el mensaje de error SQL0464W. El valor de DB2_RETURN_STATUS se recupera para procedimientos que devuelven conjuntos de resultados, pero siempre devuelve el valor cero (0), independientemente del valor de retorno real del procedimiento.

Limitaciones

El derivador SQL Server no puede llamar a un procedimiento en estas situaciones:

- Cuando ya se ha llamado a otro procedimiento.
- Cuando se ejecuta otra sentencia durante una única conexión.

Para evitar estas limitaciones, puede definir un procedimiento federado en otro servidor. En este ejemplo, las sentencias siguientes son satisfactorias si el apodo *sql_nick* y el procedimiento *sql_proc* están definidos en servidores distintos. Si el apodo y el procedimiento están definidos en el mismo servidor, las sentencias fallan.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM sql_nick OPEN clientcur; CALL sql_proc();
```

Ejemplo

Este parámetro muestra cómo utilizar la sentencia CREATE PROCEDURE para crear un procedimiento almacenado para un procedimiento de origen de datos en Microsoft SQL Server. Debe tener en cuenta que Microsoft SQL Server no da soporte a la cláusula UNIQUE ID ni al valor source_package_name de la cláusula 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

Necesario. Especifica el nombre del procedimiento federado.

SOURCE BHATIA.PROC1_MSSQL

Necesario. Especifica el nombre del esquema y el procedimiento en el origen de datos.

FOR SERVER MSSQL_SERVER

Necesario. Especifica una definición de servidor en la que se crea el procedimiento federado.

NUMBER OF PARAMETERS 5

Especifica el número total de parámetros IN y OUTPUT que se utilizan en el procedimiento de origen de datos.

SOURCE BHATIA.PROC1_SQL

Especifica el esquema y el nombre para el procedimiento de origen de datos. El formato de este nombre de dos partes es *nombre_esquema_origen.nombre_procedimiento_origen*.

SPECIFIC MYPROC1

Especifica un nombre exclusivo para el procedimiento federado que se está creando. Este parámetro sólo se utiliza para procedimientos federados y no está asociado con procedimientos de orígenes de datos. Si no especifica ningún nombre exclusivo, el gestor de bases de datos federado genera un nombre.

WITH RETURN TO CLIENT ALL

Especifica que el conjunto de resultados se devuelve a la aplicación cliente. La federación devuelve como máximo un conjunto de resultados. Si este parámetro no se especifica, el valor predeterminado es WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indica el nivel de acceso a datos para las sentencias SQL que se incluyen en el procedimiento federado. Si la cláusula especificada no coincide con el procedimiento de origen de datos, se devuelve un mensaje de error. Si esta cláusula no se especifica, se utiliza la cláusula para el procedimiento de origen de datos.

DETERMINISTIC

Especifica si el procedimiento federado siempre devuelve los mismos resultados para un conjunto de valores de argumento proporcionado. Este parámetro puede mejorar el rendimiento de la interacción entre el servidor federado y el origen de datos.

EXTERNAL ACTION

Especifica si el procedimiento federado lleva a cabo una acción que cambia el estado de un objeto que no se gestiona a través del gestor de bases de datos.

Procedimientos federados y Oracle

Puede crear un procedimiento federado y una correlación de funciones para la misma función de Oracle.

Utilice una correlación de funciones si necesita utilizar la función de Oracle en SQL como función escalar. Utilice un procedimiento federado si necesita utilizar la función de Oracle en sentencias CALL.

Oracle sólo devuelve un valor para las funciones. El valor de retorno para el procedimiento federado aparece al principio de la lista de parámetros como parámetro OUT adicional. El nombre del parámetro siempre es DEFAULT. Cuando especifique la cláusula NUMBER OF PARAMETERS en la sentencia CREATE PROCEDURE (fuente), no cuente los valores de retorno.

Para algunos tipos de datos de Oracle, la información acerca de la precisión, la longitud y la escala no se almacena en el catálogo de Oracle cuando se declaran los parámetros de un procedimiento. Cuando se crea un procedimiento federado, la información para el procedimiento de Oracle se obtiene del catálogo de Oracle. Dado que la información acerca de la precisión, la longitud y la escala no se almacena en el catálogo de Oracle, los procedimientos almacenados se comportan de este modo:

- Utilizan la longitud máxima para los tipos de datos de parámetro.
- Correlacionan los tipos de datos NUMBER de Oracle con el tipo de datos DOUBLE federado. Esta correlación puede cambiarse sustituyendo la correlación de tipos de datos directa predeterminada para Oracle NUMBER.

Consejo: La sustitución de las correlaciones de tipos de datos directas predeterminadas afecta a otras operaciones DDL federadas, como por ejemplo, CREATE NICKNAME. Por consiguiente, antes de crear el procedimiento, debe cambiar la correlación de tipos. Cree el procedimiento para el procedimiento de Oracle con la nueva correlación de tipos y luego elimine (DROP) la nueva correlación de tipos. Los apodos y procedimientos subsiguientes que cree utilizarán la correlación de tipos predeterminada.

Ejemplo

Este ejemplo muestra cómo utilizar la sentencia CREATE PROCEDURE para crear un procedimiento federado para un procedimiento de origen de datos en 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

Necesario. Especifica el nombre del procedimiento federado.

SOURCE ZELLER_SCHEMA.ORACLE_PKG9.PROC2

Necesario. Especifica el esquema, el paquete y el nombre para el procedimiento o la función de Oracle. Si el procedimiento o la función de Oracle se encuentran en un paquete, debe especificar el nombre de tres partes en la sentencia CREATE PROCEDURE. El formato para este nombre de tres partes es:

nombre_esquema_origen.nombre_paquete_origen.nombre_procedimiento_origen. Si el procedimiento o la función de Oracle no se encuentran en un paquete, debe especificar un nombre de dos partes en la sentencia CREATE PROCEDURE. El formato de este nombre de dos partes es *nombre_esquema_origen.nombre_procedimiento_origen*.

NUMBER OF PARAMETERS 5

Especifica el número total de parámetros IN, OUT e INOUT utilizados por el procedimiento de Oracle. Utilice este parámetro si tiene más de un procedimiento con el mismo nombre de esquema y nombre de procedimiento. Por ejemplo, si el esquema es ZELLER y tiene un procedimiento PROC1 con dos parámetros y otro procedimiento PROC1 con tres parámetros, el nombre para ambos procedimientos es ZELLER.PROC1. El valor para NUMBER OF PARAMETERS en el procedimiento de origen de datos indica a qué procedimiento se hace referencia en la sentencia CREATE PROCEDURE. Los parámetros REFCURSOR de Oracle deben incluirse en el recuento de NUMBER OF PARAMETERS.

UNIQUE_ID '2'

Especifica el identificador exclusivo para el procedimiento de Oracle. Utilice el parámetro UNIQUE_ID únicamente si el nombre de esquema, el nombre de procedimiento y el número de parámetros no identifican de forma unívoca un procedimiento de Oracle. El valor de UNIQUE ID es el valor de la columna ALL_ARGUMENTS.OVERLOAD del catálogo del sistema de Oracle. Si no especifica el parámetro UNIQUE ID, el servidor federado detecta los procedimientos con sobrecarga y devuelve un error. Esta opción sólo debe utilizarse con procedimientos de Oracle.

FOR SERVER ORA_SERVER

Necesario. Especifica una definición de servidor en la que se crea el procedimiento federado.

SPECIFIC MYPROC1

Especifica un nombre exclusivo para el procedimiento federado que se está creando. Este parámetro sólo se utiliza para procedimientos federados y no está asociado con procedimientos de orígenes de datos. Si no especifica ningún nombre exclusivo, el gestor de bases de datos federado genera un nombre. Este parámetro es opcional.

WITH RETURN TO CLIENT ALL

Especifica que el conjunto de resultados se devuelve a la aplicación cliente. La federación devuelve como máximo un conjunto de resultados. Si este parámetro no se especifica, el valor predeterminado es WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indica el nivel de acceso a datos para las sentencias SQL que se incluyen en el procedimiento federado. Si la cláusula especificada no coincide con el procedimiento de Oracle, se devuelve un mensaje de error. Si esta cláusula no se especifica, se utiliza la cláusula para el procedimiento de Oracle.

DETERMINISTIC

Especifica si el procedimiento federado siempre devuelve los mismos resultados para un conjunto de valores de argumento proporcionado. Este parámetro puede mejorar el rendimiento de la interacción entre el servidor federado y el origen de datos.

NO EXTERNAL ACTION

Especifica si el procedimiento federado lleva a cabo una acción que cambia el estado de un objeto que no se gestiona a través del gestor de bases de datos.

Procedimientos con sobrecarga en sistemas federados

Los procedimientos con *sobrecarga* son procedimientos que tienen nombres y esquemas idénticos. Los procedimientos con sobrecarga pueden tener un número distinto de parámetros o un número distintos de firmas de parámetros. El objetivo de sobrecargar un procedimiento es crear versiones similares de un procedimiento.

El servidor federado sólo permite procedimientos con sobrecarga si cada procedimiento tiene un número de parámetros distinto.

Oracle permite procedimientos con sobrecarga si cada procedimiento tiene un número de parámetros distinto o si el tipo de los parámetros es distinto. Puede crear procedimientos federados para procedimientos con sobrecarga de Oracle.

Para diferenciar los procedimientos que utilizan un nombre, un nombre de esquema y un número de parámetros idéntico, debe especificar el valor de UNIQUE ID cuando cree el procedimiento federado.

Existen dos formas de determinar el valor de UNIQUE ID: mediante la característica de descubrimiento del Centro de control y mediante una consulta al catálogo de Oracle.

Uso de la característica de descubrimiento en el Centro de control

Cuando se crea un procedimiento federado en el Centro de control, la característica de descubrimiento determina si el procedimiento de Oracle está sobrecargado. El valor con sobrecarga se visualiza en el campo UNIQUE ID para cada procedimiento de Oracle.

Consulta del catálogo de Oracle

Para determinar el valor de UNIQUE ID, puede realizar una consulta en la columna OVERLOAD del catálogo SYS.ALL_ARGUMENTS de Oracle. El valor de UNIQUE ID es un literal de carácter que contiene un número, como por ejemplo, 1. Puede utilizar el modo de contraseña (passthru) en el servidor federado o consultar el cliente Oracle directamente.

Por ejemplo, para consultar el catálogo de Oracle y visualizar la signatura y la columna de sobrecarga que empieza por HJZ, utilice la siguiente sentencia SELECT:

```
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 siguiente salida de la consulta anterior muestra que el paquete HJZ_PACK1 contiene tres procedimientos que utilizan el nombre HJZTEST1. Para determinar el número de procedimientos, debe consultar las columnas OBJECT_NAME y OVERLOAD. El primer procedimiento tiene un parámetro IN con un tipo de datos de número. El primer procedimiento tiene un parámetro IN con un tipo de datos de carácter. El tercer procedimiento tiene un parámetro OUT con un tipo de datos de carácter y un parámetro IN con un tipo de datos de número. La salida muestra que hay dos procedimientos en el paquete HJZ_PACK1 que utilizan el nombre HJZTEST3. También existe un procedimiento con el nombre HJZTEST1 que no se encuentra en el paquete. Este último procedimiento tiene un parámetro IN que utiliza un tipo de datos de número.

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 record(s) selected.

Para crear un procedimiento federado para el segundo procedimiento con sobrecarga con un parámetro IN con el tipo de datos CHAR, emita la siguiente sentencia CREATE PROCEDURE:

```
CREATE PROCEDURE HJZTEST1 SOURCE J15USER1.HJZ_PACK1.HJZTEST1
NUMBER OF PARAMETERS 1 UNIQUE ID '2'
FOR SERVER ORA_SERVER WITH RETURN TO CLIENT ALL;
```

Importante: En el ejemplo anterior, la cláusula NUMBER OF PARAMETERS no identifica el procedimiento de forma unívoca. Hay dos procedimientos en la tabla con el nombre HJZTEST1 y cada uno de los procedimientos tiene un parámetro. Debe especificar la cláusula UNIQUE ID para indicar el procedimiento con sobrecarga que desea utilizar. Utilice el valor de la columna OVERLOAD como valor para la cláusula UNIQUE_ID. Cuando se especifica la cláusula UNIQUE ID, la cláusula NUMBER OF PARAMETERS es opcional. Utilice la cláusula NUMBER OF PARAMETERS para validar si el procedimiento de origen de datos tiene el número de parámetros esperado.

Procedimientos federados y Sybase

Antes de crear un procedimiento federado, revise qué parámetros pueden utilizarse, cómo se devuelven los conjuntos de resultados y qué limitaciones existen.

Parámetros

Los procedimientos de Sybase utilizan los parámetros INPUT y OUTPUT. El derivador Sybase correlaciona un parámetro INPUT de Sybase con un parámetro IN federado, y un parámetro OUTPUT de Sybase, con un parámetro INOUT federado. Si bien es posible utilizar parámetros opcionales en un procedimiento de Sybase, no es posible utilizar parámetros opcionales en un procedimiento federado. Por consiguiente, cuando emita una sentencia CALL, debe especificar todos los parámetros.

Para Sybase versión 12.0, todos los parámetros son parámetros de entrada. Los procedimientos almacenados no pueden devolver un valor de parámetro de salida. Se trata de una limitación del catálogo de Sybase y no es válida para versiones posteriores de Sybase.

Conjuntos de resultados

Para los procedimientos Sybase que devuelven un parámetro de salida y un conjunto de resultados, el conjunto de resultados se descarta. Si el procedimiento Sybase devuelve un conjunto de resultados y un valor de retorno, se proporciona el valor de retorno 0, independientemente del valor de retorno real del procedimiento de datos de origen. No recibirá ningún mensaje de aviso en ninguna de estas situaciones.

No obstante, los conjuntos de resultados se descartan y se devuelve el mensaje SQL0464W cuando se cumplen las dos condiciones siguientes:

- El procedimiento federado se define para un procedimiento de Sybase que devuelve conjuntos de resultados.
- El procedimiento federado se invoca en una función desencadenante o en una función definida por el usuario.

Limitaciones

El derivador Sybase no puede llamar a un procedimiento en estas situaciones:

- Cuando ya se ha llamado a otro procedimiento.
- Cuando se ejecuta otra sentencia durante una única conexión.

Para evitar estas limitaciones, puede definir un procedimiento federado en otro servidor. En este ejemplo de Sybase, las sentencias siguientes son satisfactorias si el apodo *syb_nick* y el procedimiento *syb_proc* están definidos en servidores distintos. Si el apodo y el procedimiento están definidos en el mismo servidor, las sentencias fallan.

```
DECLARE clientcur CURSOR FOR SELECT colsm1,coldec,colvch,coltsp  
FROM syb_nick OPEN clientcur; CALL syb_proc();
```


Ejemplo

Este ejemplo muestra cómo utilizar la sentencia CREATE PROCEDURE para crear un procedimiento federado para un procedimiento de origen de datos en Sybase. Debe tener en cuenta que Sybase no da soporte a la cláusula UNIQUE ID de la sentencia 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

Necesario. Especifica el nombre del procedimiento federado.

SOURCE BHATIA.PROC1_SYBASE

Necesario. Especifica el esquema y el nombre para el procedimiento de Sybase. Para procedimientos Sybase, debe especificar un nombre de dos partes en la sentencia CREATE PROCEDURE. El formato de este nombre de dos partes es *nombre_esquema_origen.nombre_procedimiento_origen*.

NUMBER OF PARAMETERS 3

Especifica el número total de parámetros IN, OUT e INOUT utilizados por el procedimiento de Sybase. Utilice este parámetro si tiene más de un procedimiento con el mismo nombre de esquema y nombre de procedimiento. Por ejemplo, si el esquema es BHATIA y tiene un procedimiento PROC1 con tres parámetros y otro procedimiento PROC1 con un parámetro, el nombre para ambos procedimientos es BHATIA.PROC1. El valor para NUMBER OF PARAMETERS en el procedimiento de origen de datos indica a qué procedimiento se hace referencia en la sentencia CREATE PROCEDURE.

FOR SERVER SYBASE_SERVER

Necesario. Especifica una definición de servidor en la que se crea el procedimiento federado.

SPECIFIC MYPROC1

Especifica un nombre exclusivo para el procedimiento federado que se está creando. Este parámetro sólo se utiliza para procedimientos federados y no está asociado con procedimientos de orígenes de datos. Si no especifica ningún nombre exclusivo, el gestor de bases de datos federado genera un nombre.

WITH RETURN TO CLIENT ALL

Especifica que el conjunto de resultados se devuelve a la aplicación cliente. La federación devuelve como máximo un conjunto de resultados. Si este parámetro no se especifica, el valor predeterminado es WITH RETURN TO CALLER ALL.

MODIFIES SQL DATA

Indica el nivel de acceso a datos para las sentencias SQL que se incluyen en el procedimiento federado. Si la cláusula especificada no coincide con el procedimiento de Sybase, se devuelve un mensaje de error. Si esta cláusula no se especifica, se utiliza la cláusula para el procedimiento de Sybase.

DETERMINISTIC

Especifica si el procedimiento federado siempre devuelve los mismos resultados para un conjunto de valores de argumento proporcionado. Este parámetro puede mejorar el rendimiento de la interacción entre el servidor federado y el origen de datos.

EXTERNAL ACTION

Especifica si el procedimiento federado lleva a cabo una acción que cambia el estado de un objeto que no se gestiona a través del gestor de bases de datos.

Creación de procedimientos federados

Debe crear un procedimiento almacenado para cada procedimiento de origen de datos que desee invocar desde el servidor federado.

Antes de empezar

- El procedimiento de origen de datos ya debe existir.
- El derivador para el origen de datos debe estar registrado y se debe haber creado la definición de servidor.
- Deben haberse creado las correlaciones de usuario necesarias.
- El ID de autorización de la sentencia debe tener al menos uno de los privilegios siguientes:
 - Privilegio IMPLICIT_SCHEMA sobre la base de datos, si el nombre de esquema del procedimiento no hace referencia a un esquema existente.
 - Privilegio CREATEIN sobre el esquema, si el nombre de esquema del procedimiento hace referencia a un esquema existente.
 - Autorización SYSADM o DBADM.
- El ID de autorización del origen de datos también debe tener el privilegio para seleccionar la descripción del procedimiento de la tablas de catálogo remotas.

Acerca de esta tarea

Para crear un procedimiento federado, utilice la sentencia SQL CREATE PROCEDURE (fuente) o el Centro de control. Si utiliza el Centro de control, seleccione un procedimiento de origen de datos, ya que el Centro de control recuperará automáticamente la información necesaria para crear el procedimiento federado.

Para utilizar el Centro de control:

1. Pulse con el botón derecho en la carpeta **Procedimientos almacenados federados** y, a continuación, pulse **Crear**. La carpeta **Procedimientos almacenados federados** se encuentra en el derivador y en la definición de servidor que ha registrado para el origen de datos.
2. En la ventana **Crear procedimiento almacenado federado**, pulse **Descubrir**.
3. En la ventana **Descubrir**, especifique los criterios que filtran las búsquedas por esquema, nombre de paquete, nombre de procedimiento o nombre de función. Utilice los operadores BETWEEN, IN, NOT IN y LIKE para especificar criterios, e indique los valores entre comillas simples. Utilice el botón **Contar** de la ventana **Descubrir** para determinar cuántos procedimientos se devolverán. Si el número es elevado, especifique más criterios para limitar la búsqueda.
4. Opcional: En la ventana **Crear procedimiento almacenado federado**, active el recuadro de selección que se encuentra junto al procedimiento que desea crear y pulse **Propiedades** para verificar la información y asegurarse de que todos los campos necesarios tienen un valor antes de crear el procedimiento federado.
5. En la ventana **Crear procedimiento almacenado federado**, active el recuadro de selección que se encuentra junto a uno o más procedimientos que desee crear y pulse **Aceptar**.

Cómo otorgar o revocar autorizaciones para llamar a procedimientos federados

El administrador de la base de datos federada debe otorgar a otros usuarios la autorización necesaria para llamar a los procedimientos federados.

Antes de empezar

El usuario que llama al procedimiento federado debe tener una correlación de usuario válida del servidor federado al origen de datos. El ID de usuario remoto debe tener una autorización en el origen de datos que sea equivalente a la autorización EXECUTE en el servidor federado. Es posible otorgar autorización EXECUTE a un usuario en el procedimiento federado. No obstante, si la autorización del usuario en el origen de datos no es equivalente a la autorización EXECUTE en el servidor federado, las llamadas al procedimiento de origen de datos fallarán.

El ID de autorización para la sentencia GRANT debe tener como mínimo una de estas autorizaciones:

- WITH GRANT OPTION para EXECUTE en el procedimiento almacenado
- Autorización SYSADM o DBADM

Procedimiento

Para otorgar o revocar la autorización para llamar a procedimientos almacenados:

Especifique los privilegios de autorización desde el Centro de control o la línea de mandatos:

Método	Descripción
Centro de control	<ol style="list-style-type: none"> 1. Pulse con el botón derecho en el nombre del procedimiento almacenado y seleccione Privilegios. 2. Seleccione el usuario o grupo para el que desea definir privilegios. Para añadir un nuevo usuario, pulse Añadir usuario. Para añadir un nuevo grupo, seleccione la pestaña Grupo y pulse Añadir grupo. 3. En el recuadro desplegable Privilegios: EXECUTE: <ul style="list-style-type: none"> • Seleccione Sí para otorgar sólo el privilegio EXECUTE. • Seleccione Otorgar para otorgar los privilegios EXECUTE y WITH GRANT OPTION. • Seleccione No para eliminar el privilegio EXECUTE del usuario o grupo. 4. Puede otorgar o revocar privilegios a varios usuarios al mismo tiempo. Seleccione los usuarios de la lista y pulse Otorgar a todos o Revocar a todos para otorgar o revocar el privilegio EXECUTE a todos los usuarios seleccionados. 5. Pulse Aceptar.

Método	Descripción
<p>Línea de mandatos</p>	<p>Especifique los privilegios en la sentencia GRANT.</p> <p>Ejemplo 1: Para otorgar el privilegio EXECUTE sobre todos los procedimientos del esquema BHATIA, incluyendo los procedimientos que se creen en un futuro, a los usuarios del grupo HR_DEPT, utilice esta sentencia: GRANT EXECUTE ON PROCEDURE BHATIA.* TO HR_DEPT</p> <p>Ejemplo 2: Para otorgar el privilegio EXECUTE sobre el procedimiento PROC1 al usuario ZELLER y conceder a esta persona la capacidad de otorgar el privilegio EXECUTE sobre este procedimiento a otros usuarios, utilice la siguiente sentencia: GRANT EXECUTE ON PROCEDURE PROC1 TO ZELLER WITH GRANT OPTION</p> <p>Ejemplo 3: Para otorgar el privilegio EXECUTE al usuario ERFAN en el procedimiento PROC2 creado con el nombre específico MY_PROC2, utilice esta sentencia: GRANT EXECUTE ON SPECIFIC PROCEDURE MY_PROC2 TO ERFAN</p>

Visualización de información de parámetros y llamada a procedimientos federados

Utilice el Centro de control o consulte la vista de catálogo SYSCAT.ROUTINESFEDERATED para visualizar información sobre parámetros. A continuación, utilice la sentencia CALL para llamar a un procedimiento federado.

Antes de empezar

El usuario que llama a un procedimiento federado debe tener el privilegio EXECUTE en el origen de datos y una correlación de usuario válida con un ID de usuario de origen de datos remoto que tenga privilegio para acceder a tablas del origen de datos.

Para visualizar información sobre parámetros y llamar a un procedimiento federado, utilice uno de estos métodos:

Método	Procedimiento
Centro de control	<ol style="list-style-type: none"> 1. Expanda las carpetas Objetos federados, Definiciones de servidor y Procedimientos almacenados federados del árbol de objetos. 2. Seleccione el nombre del procedimiento federado para visualizar información como el tipo de datos y el modo de cada parámetro. 3. Pulse HerramientasEditor de mandatos para abrir el Editor de mandatos y emitir una sentencia CALL.
Línea de mandatos	<ol style="list-style-type: none"> 1. Utilice la sentencia SELECT para consultar la vista SYSCAT.ROUTINEPARMS en el catálogo de bases de datos federadas. 2. Emita una sentencia CALL.

Este ejemplo utiliza una sentencia SELECT para obtener información sobre el procedimiento federado. Por ejemplo, si el procedimiento federado FEDPROC1 se encuentra en el BOB de esquema federado, emita esta sentencia SELECT:

```
SELECT ordinal, char(parmname,30)
AS name,
rowtype, char(tyename,30) AS type
FROM syscat.routineparms
WHERE routinename='FEDPROC1' AND
routineschema = 'BOB'
ORDER BY ordinal;
```

El resultado de la consulta enumera los parámetros:

```
ORDINAL NAME ROWTYPE TYPE
-----
1      P1  P      INTEGER
2      P2  O      VARCHAR
```

El tipo de fila P indica un parámetro de entrada. El tipo de fila O indica un parámetro de salida. En este ejemplo no se muestra el tipo de fila B, que indica un parámetro INOUT.

Modificación o eliminación de procedimientos federados

Puede modificar un procedimiento almacenado existente cambiando el tipo de datos de uno o más parámetros del procedimiento federado. No es posible modificar el procedimiento federado directamente para realizar otros cambios.

Antes de empezar

El ID de autorización para la sentencia DROP PROCEDURE debe tener una de las autorizaciones siguientes:

- Autorización SYSADM o DBADM
- Privilegio DROPIN en el esquema para el procedimiento almacenado
- Definidor del procedimiento tal como se encuentra registrado en la columna DEFINER de la vista de catálogo para el procedimiento almacenado
- Privilegio CONTROL en el procedimiento almacenado

Acerca de esta tarea

Además de cambiar el tipo de datos de un parámetro, es posible que deba realizar otros cambios en un procedimiento federado. Por ejemplo, es posible que deba cambiar un procedimiento federado si el procedimiento remoto cambia. En este caso, no puede modificar directamente el procedimiento federado. Primero debe eliminar el procedimiento y luego volver a crearlo con los nuevos valores. En caso contrario, deberá sustituir el procedimiento existente utilizando la sentencia CREATE OR REPLACE PROCEDURE.

Al eliminar un procedimiento federado, el procedimiento se suprime del catálogo del sistema de la base de datos federada, pero el procedimiento del origen de datos no se ve afectado. Debe tener en cuenta que, al eliminar un procedimiento federado, las aplicaciones que dependan de los procedimientos eliminados pueden verse negativamente afectadas.

Puede utilizar el Centro de control o la línea de mandatos para eliminar un procedimiento federado.

Procedimiento

Para eliminar un procedimiento federado, utilice uno de estos métodos:

Método	Procedimiento
Centro de control	<ol style="list-style-type: none">1. Expanda las carpetas Objetos federados, Definiciones de servidor y Procedimientos almacenados federados del árbol de objetos.2. Pulse con el botón derecho en el procedimiento almacenado que desea eliminar y seleccione Eliminar.
Línea de mandatos	Utilice la sentencia DROP. Por ejemplo: DROP PROCEDURE <i>nombre_procedimiento_almacenado</i>

Unión de conjuntos de resultados de procedimientos federados

Es posible unir los conjuntos de resultados devueltos por un procedimiento federado con el mandato DB2FEDGENTF.

Antes de empezar

- Para utilizar el parámetro **-create** del mandato DB2FEDGENTF, debe tener autorización DBADM en la base de datos federada o una combinación de las autorizaciones siguientes en dicha base de datos:
 1. Debe tener una de las autorizaciones siguientes:
 - CREATE_EXTERNAL_ROUTINE
 - CREATETAB
 2. También debe tener una de las autorizaciones o uno de los privilegios siguientes:
 - Privilegio de escritura para el directorio *dir_instalación/sqllib/function*, donde *dir_instalación* es el directorio en el que se halla instalado el servidor federado.
 - Autorización IMPLICIT_SCHEMA si el nombre de esquema implícito o explícito de la función no existe.

- El privilegio CREATEIN en el esquema si el nombre del esquema de la función existe.
- Para utilizar el parámetro **-drop** del mandato DB2FEDGENTF, debe tener como mínimo uno de las autorizaciones o uno de los privilegios siguientes en la base de datos federada:
 - Privilegio DROPIN sobre el esquema correspondiente al objeto.
 - Debe ser el propietario (OWNER) del objeto.
 - Debe tener autorización DBADM.

Acerca de esta tarea

Los conjuntos de resultados que devuelven los procedimientos federados se unen para unir los datos de las tablas locales y remotas, especialmente en sistemas que sólo permiten el acceso a través de procedimientos federados.

Procedimiento

Para unir los conjuntos de resultados de procedimientos federados:

1. Cree y registre una función de tabla con el parámetro **-create** del mandato DB2FEDGENTF, por ejemplo:


```
DB2FEDGENTF -db sample -u user1 -p password1
  -create
    -stpn S1_INVENTORY
    -tfn S1_INVTRY_TF
    -c 'PART_NUM CHAR(10), S1_QTY INT'
```
2. Utilice una unión para combinar los datos de los conjuntos de resultados de procedimientos federados. Puede unir el conjunto de resultados de un procedimiento federado con una tabla local o unir los conjuntos de resultados de dos procedimientos federados.

Ejemplos

En los ejemplos siguientes, los procedimientos almacenados denominados ProINVENTORY y ProINVENTORY2 existen y cada uno devuelve el inventario de dos proveedores como conjuntos de resultados. La tabla PARTS también existe y contiene el inventario del fabricante con los nombres y tipos de columna siguientes:

Tabla 6. Columnas de la tabla PARTS

Nombre de columna	Tipo de columna
PART_NUM	CHAR(10)
PART_DESC	VARCHAR(400)
INVENTORY	INT

Ejemplo de unión de conjuntos de resultados con tablas locales

En este ejemplo, el inventario entre el proveedor y el fabricante se combinan para determinar el inventario total disponible:

1. Utilice la sentencia CREATE PROCEDURE para crear el procedimiento federado S1_INVENTORY a partir del procedimiento almacenado ProINVENTORY:


```
CREATE PROCEDURE S1_INVENTORY
  SOURCE ProINVENTORY
  FOR SERVER ORA1;
```


2. Utilice el mandato DB2FEDGENTF para crear la tabla de función S1_INVTRY_TF que incluye el conjunto de resultados PART_NUM y S1_QTY:

```
DB2FEDGENTF -db sample
-u user1 -p password1
-create
  -stpn S1_INVENTORY
  -tfn S1_INVTRY_TF
  -c 'PART_NUM CHAR(10), S1_QTY INT'
```

3. Utilice una unión para combinar los datos de la tabla PARTS con el conjunto de resultados de la función de tabla 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
```

Ejemplo de unión de dos conjuntos de resultados

En este ejemplo, el fabricante combina el inventario de dos proveedores para determinar su inventario disponible total:

1. Cree la función de tabla S1_INVTRY_TF para el primer proveedor:

- a. Utilice la sentencia CREATE PROCEDURE para crear el procedimiento federado S1_INVENTORY a partir del procedimiento almacenado ProINVENTORY:

```
CREATE PROCEDURE S1_INVENTORY
SOURCE ProINVENTORY
FOR SERVER ORA1;
```

- b. Utilice el mandato DB2FEDGENTF para crear la tabla de función S1_INVTRY_TF que incluye el conjunto de resultados PART_NUM y S1_QTY:

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S1_INVENTORY
  -tfn S1_INVTRY_TF
  -c 'PART_NUM CHAR(10), S1_QTY INT'
```

2. Cree la función de tabla S2_INVTRY_TF para el segundo proveedor:

- a. Utilice la sentencia CREATE PROCEDURE para crear el procedimiento federado S2_INVENTORY a partir del procedimiento almacenado ProINVENTORY2:

```
CREATE PROCEDURE S2_INVENTORY
SOURCE ProINVENTORY2
FOR SERVER SYBA1;
```

- b. Utilice el mandato DB2FEDGENTF para crear la tabla de función S2_INVTRY_TF que incluye el conjunto de resultados PART_NUM y S2_QTY:

```
DB2FEDGENTF -db sample -u user1 -p password1
-create
  -stpn S2_INVENTORY
  -tfn S2_INVTRY_TF
  -c 'PART_NUM CHAR(10), S2_QTY INT'
```

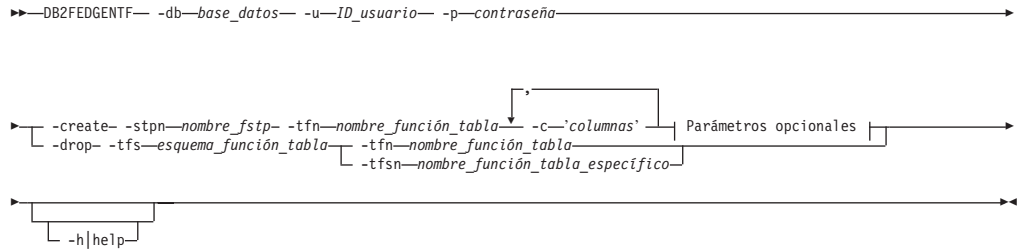
3. Utilice una unión para combinar los conjuntos de resultados de las funciones de tabla S1_INVTRY_TF y S2_INVTRY_TF:

```
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
```

Mandato DB2FEDGENTF

Crea y registra las funciones de tabla que devuelven conjuntos de resultados de procedimientos almacenados.

Sintaxis



Parámetros opcionales:



Parámetros

-db *base_datos*

Especifica el nombre de la base de datos con la que se desea establecer conexión.

-u *ID_usuario*

Especifica el ID de usuario de la base de datos federada.

-p *contraseña*

Especifica la contraseña del ID de usuario.

-create

Crea y registra una función de tabla en el esquema actual si no se especifica el parámetro **-tfs**. La función de tabla devuelve las columnas especificadas del conjunto de resultados del procedimiento federado.

-stpn *nombre_fstp*

Especifica el nombre del procedimiento federado.

-tfn *nombre_función_tabla*

Especifica el nombre de la función de tabla. Si no se especifica el nombre de la función de tabla, se utiliza el nombre del procedimiento federado.

-c *'columnas'*

Especifica una lista separada por comas que incluye pares de nombre de columna y tipo de columna en la signatura del conjunto de resultados devuelto por el procedimiento federado.

Ejemplo

Los nombres de columna son PID, PRICE y QTY, y los tipos de columna son CHAR(10), DOUBLE y INT:

```
'PID CHAR(10), PRICE DOUBLE, QTY INT'
```

-stps *esquema_fstp*

Especifica el esquema del procedimiento almacenado. Este parámetro es

opcional. Si no se especifica ningún nombre, se utiliza el esquema SQL predeterminado según está definido en el registro especial CURRENT SCHEMA.

-stpc *número_ parámetros_ Fstp*

Especifica el número de entradas para el procedimiento federado. Este parámetro es opcional. Si no se especifica el número de entradas, el procedimiento federado se determina a partir del nombre del procedimiento federado especificado. Si los procedimientos federados están sobrecargados, se recibe un error.

-tfs *esquema_ función_ tabla*

Especifica el esquema de la función de tabla. Este parámetro es opcional. Si no se especifica ningún esquema de la función de tabla, se utiliza el esquema SQL predeterminado según está definido en el registro especial CURRENT SCHEMA.

- drop

Elimina la función de tabla especificada. La descripción del catálogo también se elimina y todos los paquetes que hacen referencia a la función de tabla especificada dejan de ser válidos.

-tfs *esquema_ función_ tabla*

Especifica el esquema de la función de tabla que se desea eliminar. Si no se especifica ningún esquema de la función de tabla, se utiliza el esquema SQL predeterminado según está definido en el registro especial CURRENT SCHEMA.

-tfn *nombre_ función_ tabla*

Especifica el nombre de la función de tabla que se desea eliminar.

-tfsn *nombre_ función_ tabla_ específico*

Para funciones sobrecargadas, especifica el nombre específico de la función de tabla que se desea eliminar. Este parámetro se excluye mutuamente con el parámetro **-tfn** *nombre_ función_ tabla*. No es necesario especificar esta opción si la función de tabla se identifica de forma unívoca mediante su nombre y esquema. Este parámetro es opcional.

-h | help

Proporciona información de uso para el mandato DB2FEDGENTF.

Ejemplos

En este ejemplo, el mandato DB2FEDGENTF devuelve el procedimiento federado EAST_INVTRY para crear la función de tabla S1_INVTRY_TF que devuelve las columnas PRODID, PRICE y QTY:

```
DB2FEDGENTF -db sample -u user1 -p password1
  -create
    -stpn EAST_INVTRY
    -tfn E_INVTRY_TF
    -c 'PRODID INT, PRICE DOUBLE, QTY INT'
```

Resolución de problemas de procedimientos federados

Si detecta problemas con los procedimientos federados, existen varios métodos para resolverlos.

Las consultas y herramientas de diagnóstico siguientes le ayudarán a ver información sobre los procedimientos federados. Esta información le ayudará a resolver problemas con los procedimientos federados.

Verificación de la información del procedimiento de origen de datos

Si se devuelve el error SQL1253N al emitir una sentencia CREATE PROCEDURE, puede emitir las consultas siguientes en las tablas del catálogo del origen de datos para verificar la información sobre el procedimiento de origen de datos. El error SQL1253N indica que el procedimiento de origen de datos especificado en la sentencia CREATE PROCEDURE (fuente) no se ha encontrado en el origen de datos. Puede consultar directamente el servidor Oracle o utilizar una sesión de paso a través para consultar el servidor Oracle.

Para procedimientos en DB2 for Linux, UNIX y 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
```

Para procedimientos en 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
```

Para procedimientos en 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'
```

Para Microsoft SQL Server

```
SELECT id
FROM dbo.sysobjects
WHERE id = object_id AND
      (TYPE = 'P' or TYPE = 'X')
```

Para procedimientos Oracle que se encuentran en un paquete:

```
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

```

Para procedimientos Oracle que no se encuentran en un paquete:

```

SELECT object_name, object_type, status
FROM sys.all_objects
WHERE owner = '' AND
object_name = '' AND
object_type IN ('PROCEDURE', 'FUNCTION')

```

For Sybase procedures:

```

SELECT id
FROM dbo.sysobjects
WHERE id = object_id('.') AND
(TYPE = 'P' OR TYPE = 'XP')

```

Herramientas de diagnóstico

Utilice el programa de utilidad Explain, el mandato DESCRIBE o el mandato db2audit para diagnosticar los problemas con los procedimientos almacenados.

Por ejemplo, el procedimiento FED_PROC1 tiene tres parámetros OUTPUT. Para utilizar el mandato DESCRIBE en el procedimiento FED_PROC1, emita este mandato:

```
DESCRIBE CALL FED_PROC1(?,?,?);
```

Supervisor del sistema

Los elementos del supervisor del sistema de la base de datos federada contienen información sobre los procedimientos federados. Los elementos del supervisor son:

- El elemento de supervisor Tiempo de procedimiento almacenado, `stored_proc_time`, que contiene el tiempo que ha tardado el origen de datos en responder a las sentencias de procedimientos federados.
- El elemento de supervisor Filas devueltas por procedimientos almacenados, `sp_rows_selected`, que contiene el número de filas que se envían desde el origen de datos al servidor federado. Puede utilizar este elemento para calcular el número promedio de filas enviadas al servidor federado desde el origen de datos para cada procedimiento federado, o para calcular el tiempo promedio para devolver una fila al servidor federado desde el origen de datos.
- El elemento de supervisor Procedimientos almacenados, `stored_procs`, que contiene un recuento del número total de procedimientos llamados por el servidor federado desde este origen de datos.

Error SQL SQL30090N con el código de retorno 21

Existen varias situaciones en las que se devolverá el error SQL30090N con el código de retorno 21. Una de las situaciones más comunes tiene lugar cuando se crea un procedimiento federado mediante un derivador con limitaciones. Los procedimientos federados sólo pueden crearse en derivadores de confianza.

Conjunto de resultados no devuelto

Es posible que un conjunto de resultados no se devuelva a un cliente o emisor de llamada por alguno de estos motivos:

- La cláusula para devolver el conjunto de resultados no se ha especificado correctamente en el procedimiento federado.
- Algunos orígenes de datos no devuelven conjuntos de resultados en el mismo orden cada vez que se llama a un procedimiento. Dado que los procedimientos federados sólo devuelven el primer conjunto de resultados, es posible que se devuelva un conjunto de resultados distinto del origen de datos cuando se llame al procedimiento federado.

Por ejemplo, existen dos procedimientos en el origen de datos, PROCEDURE A y PROCEDURE B, y éste llama al primero. Las sentencias para crear estos procedimientos son:

```
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;
```

El procedimiento federado FEDPROC1 hace referencia al origen de datos PROCEDURE B. La sentencia para el procedimiento FEDPROC1 es la siguiente:

```
CREATE PROCEDURE FEDPROC1
SOURCE newton.B
FOR SERVER s1
NUMBER OF PARAMETERS 1
WITH RETURN TO CLIENT 1;
```

Un procedimiento local llama al procedimiento federado FEDPROC1. La sentencia para el procedimiento local es:

```
CREATE PROCEDURE local (arg1 INT)
BEGIN
  CALL FEDPROC1 (arg1)
END;
```

Al emitir la sentencia CALL LOCAL(1), se devuelve el conjunto de resultados cur1 desde PROCEDURE A. El conjunto de resultados cur2 no se devuelve.

No obstante, si emite la sentencia CALL LOCAL(20), se devuelve el conjunto de resultados cur2 desde PROCEDURE B.

Sesión de paso a través (sólo Oracle)

Si crea un procedimiento de origen de datos, una función o un paquete en una sesión de paso a través, se devuelve un mensaje satisfactorio aun cuando la definición del objeto presente un error. El objeto se crea en el servidor Oracle, pero se marca con INVALID. No es posible crear procedimientos federados en objetos que no son válidos. Si intenta crear un procedimiento federado que hace referencia a un objeto Oracle marcado como INVALID, la sentencia CREATE PROCEDURE (fuente) falla.

Utilice uno de los métodos siguientes para determinar porqué un objeto no es válido:

- Utilice el mandato SHOW ERRORS en el programa de utilidad SQL*Plus desde Oracle.
- Consulte la tabla de catálogo sys.all_errors de Oracle.

Capítulo 7. Creación y modificación de tablas remotas utilizando DDL transparente

Con DDL transparente, puede utilizar procedimientos con los que está familiarizado para crear y modificar tablas remotas mediante la base de datos federada sin utilizar una sesión de paso a través.

¿Qué es el DDL transparente?

El *DDL transparente* proporciona la capacidad para crear y modificar tablas remotas mediante bases de datos federadas sin utilizar sesiones de paso a través.

Las sentencias de SQL que se utilizan con DDL transparente son CREATE TABLE, ALTER TABLE y DROP TABLE.

Una sentencia CREATE TABLE de DDL transparente crea una tabla remota en el origen de datos y un apodo para dicha tabla en el servidor federado. Correlacionará los tipos de datos de DB2 que especifique con los tipos de datos remotos mediante correlaciones de tipos de datos inversas por omisión. En general, los derivadores proporcionan las correlaciones de tipos. También pueden crearse correlaciones de tipos inversas definidas por el usuario para alterar temporalmente las correlaciones por omisión.

La ventaja de utilizar el DDL transparente consiste en que los administradores de bases de datos pueden utilizar los procedimientos que conocen para crear tanto tablas locales como remotas. El DDL transparente centraliza la administración de tablas y facilita el otorgamiento de autorizaciones.

El DDL transparente recibe soporte con los orígenes de datos siguientes:

- DB2 para z/OS
- DB2 para System i
- DB2 Database para Linux, UNIX, y Windows
- DB2 Server para VM y VSE
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle
- Sybase
- Teradata

El administrador de bases de datos puede utilizar el Centro de control de DB2 o las sentencias de DDL del procesador de línea de mandatos (CLP) de DB2 para crear las tablas. Al utilizar el DDL transparente se evita tener que aprender las distintas sintaxis de DDL que se requieren para cada origen de datos.

Antes de poder crear tablas remotas en un origen de datos mediante la base de datos federada, es necesario que configure el acceso al origen de datos:

- El derivador para dicho origen de datos debe registrarse en el catálogo global

- La definición de servidor debe crearse para el servidor en el que se ubicará la tabla remota
- Las correlaciones de usuario, si son necesarias, deben crearse entre el servidor federado y el servidor de orígenes de datos

Utilice el asistente de tablas remotas del Centro de control de DB2 para crear tablas remotas.

El ID de autorización de las sentencias de DDL transparente debe incluir al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Autoridad CREATETAB en la base de datos y privilegio USE en el espacio de tabla, así como uno de los privilegios siguientes:
 - Autoridad IMPLICIT_SCHEMA en la base de datos, si el nombre de esquema implícito o explícito de la tabla no existe
 - Privilegio CREATEIN en el esquema, si el nombre de esquema de la tabla hace referencia a un esquema existente

Para emitir sentencias de DDL transparente, su ID de autorización debe contener los privilegios necesarios en el apodo (para que el servidor federado acepte la petición), y los privilegios comparables en el origen de datos remoto (para que el origen de datos acepte la petición).

Columnas LOB remotas y DDL transparente

Especifique la longitud de una columna LOB cuando utilice DDL transparente.

Algunos orígenes de datos como, por ejemplo, Oracle e Informix no almacenan las longitudes de las columnas LOB en los catálogos de sus sistemas. Al crear un apodo en una tabla, se recupera la información procedente del catálogo del sistema de orígenes de datos, incluyendo la longitud de columna. Puesto que no existe ninguna longitud para las columnas LOB, la base de datos federada asume que dicha longitud es la longitud máxima de una columna LOB en DB2 Database para Linux, UNIX y Windows. La base de datos federada almacena la longitud máxima en el catálogo de la base de datos federada como la longitud de la columna de apodo.

No obstante, cuando cree una tabla remota mediante DDL transparente debe especificar la longitud de la columna LOB. Cuando el servidor federado crea un apodo en la tabla remota, almacena la longitud especificada en el catálogo de la base de datos federada como la longitud de la columna de apodo. La longitud máxima de una columna LOB es de 2 gigabytes.

Creación de tablas remotas y DDL transparente

Cuando se crea una tabla remota mediante la base de datos federada utilizando DDL transparente, se producen varias acciones.

Cuando se crea una tabla remota:

- Se crea automáticamente un apodo para dicha tabla remota. El apodo tiene el mismo nombre que el nombre de la tabla especificado en la sentencia CREATE TABLE. La tabla remota tendrá el mismo nombre que el nombre de tabla, a menos que se especifique otro nombre utilizando la opción REMOTE_TABNAME.

- El esquema de la tabla remota será el esquema de apodo, a menos que se especifique otro esquema utilizando la opción REMOTE_SCHEMA.
- El apodo creado mediante DDL transparente puede utilizarse igual que cualquier otro apodo. Además, podrá ALTER y DROP la tabla remota (algo que no se puede hacer con un apodo creado utilizando CREATE NICKNAME).
- Se añadirá una fila en la vista de catálogo SYSCAT.TABOPTIONS con un nombre de opción de TRANSPARENT y un valor de Y.

Creación de tablas remotas nuevas utilizando DDL transparente

Para crear tablas remotas mediante DDL transparente, puede utilizar el asistente del Centro de control de DB2 o la sentencia CREATE TABLE.

Antes de empezar

Antes de crear una tabla remota, debe configurar el servidor federado para acceder al origen de datos. Esta configuración incluye:

- Crear el derivador para ese tipo de origen de datos
- Suministrar la definición de servidor para el servidor en el que se ubicará la tabla remota
- Crear las correlaciones de usuarios entre el servidor federado y el servidor de origen de datos

Para emitir sentencias de DDL transparente, su ID de autorización debe contener los privilegios necesarios en el apodo (para que el servidor federado acepte la petición), y los privilegios comparables en el origen de datos remoto (para que el origen de datos acepte la petición).

El ID de autorización que emite las sentencias de DDL transparente debe incluir al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Autoridad CREATETAB en la base de datos y privilegio USE en el espacio de tabla, así como uno de los privilegios siguientes:
 - Autoridad IMPLICIT_SCHEMA en la base de datos, si el nombre de esquema implícito o explícito de la tabla no existe
 - Privilegio CREATEIN en el esquema, si el nombre de esquema de la tabla hace referencia a un esquema existente

Restricciones

Las restricciones siguientes son aplicables a la creación de una tabla remota mediante DDL transparente:

- No puede modificar o descartar las tablas que se hayan creado originalmente en el origen de datos remoto.
- No puede crear tablas de consultas materializadas en orígenes de datos remotas.
- Puede especificar información de columna básica en la definición de tabla, pero no podrá especificar opciones de tabla ni opciones de columna. Por ejemplo, las opciones LOB (LOGGED y COMPACT) no reciben soporte.
- No puede especificar un comentario en una columna.
- No puede generar contenidos de columna.

- Puede especificar una clave primaria, pero no podrá especificar restricciones de clave foránea o de comprobación. Las columnas utilizadas para la clave primaria deben ser NOT NULL, y no pueden incluir columnas que contengan objetos LOB.
- No puede modificar los parámetros de columnas existentes como, por ejemplo, el tipo o longitud de los datos.
- La cláusula DEFAULT de la sentencia CREATE TABLE no recibe soporte.

Acerca de esta tarea

Utilice el asistente de tablas remotas del Centro de control de DB2 para evitar la especificación de un parámetro u opción que no reciba soporte. Mediante el asistente, puede especificar las columnas seleccionándolas de una lista de columnas predefinidas o especificando los atributos para una nueva columna.

Procedimiento

Para crear una tabla remota desde el indicador de línea de mandatos, emita la sentencia CREATE TABLE con los parámetros apropiados establecidos.

Para crear una tabla remota en el Centro de control de DB2, utilice el asistente de Crear tablas remotas:

1. Expande la carpeta **Objetos de base de datos federada**.
2. Expande los objetos del derivador y de la definición de servidor para el origen de datos para la que desee crear una tabla remota.
3. Pulse con el botón derecho del ratón en la carpeta **Tablas remotas** y, a continuación, pulse **Crear**. Se iniciará el asistente Crear tablas remotas.
4. Complete los pasos del asistente.

Creación de tablas remotas nuevas utilizando DDL transparente - ejemplos

Los ejemplos siguientes ilustran qué especificar para crear tablas remotas nuevas utilizando DDL transparente y el uso de correlaciones de tipos de datos.

Cuando se crea una tabla remota utilizando DDL transparente:

- El origen de datos remoto debe dar soporte a los tipos de datos de columna y a la opción de clave primaria en la sentencia CREATE TABLE.

Ejemplo: El origen de datos remoto no da soporte a las claves primarias. Dependiendo de la forma en que el origen de datos responde a las peticiones a las que no da soporte, puede que se devuelva un error o puede que la petición sea ignorada.

- El servidor remoto debe especificarse en la cláusula OPTIONS. La cláusula OPTIONS puede utilizarse para alterar temporalmente el nombre remoto o el esquema remoto de la tabla que se está creando. La opción SQL_SUFFIX se permite al final de la sentencia CREATE TABLE. Puede especificar esta opción para que cada origen de datos relacional añada opciones específicas del origen de datos a la sentencia CREATE TABLE que se emite en el origen de datos.

Ejemplo: Desea crear la tabla EMPLOY en un servidor de Oracle. En la sentencia CREATE TABLE, utilice los tipos de datos de DB2 cuando especifique cada columna. Utilizando el CLP, la sintaxis para crear la tabla es:

```

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

El nombre del apodo asociado a la tabla.

REMOTE_SERVER 'ORASERVER'

el nombre que ha proporcionado para el servidor en la sentencia CREATE SERVER. Este valor es sensible a mayúsculas y minúsculas.

REMOTE_SCHEMA 'J15USER1'

El nombre del esquema remoto. Aunque este parámetro sea opcional, se recomienda especificar un nombre de esquema. Si no se especifica este parámetro, el esquema de apodo se utilizará para el nombre de esquema remoto. Este valor es sensible a mayúsculas y minúsculas.

REMOTE_TABNAME 'EMPLOY'

El nombre de la tabla remota. Este parámetro es opcional. Si no se especifica este parámetro, el nombre de la tabla local se utilizará para el nombre de la tabla remota. Este valor debe ser un nombre válido del origen de datos remota y no puede ser un nombre de tabla existente. Este valor es sensible a mayúsculas y minúsculas.

En el ejemplo anterior, la base de datos federada utiliza correlaciones de tipos de datos inversas para correlacionar los tipos de datos de DB2 con los tipos de datos de Oracle. En el servidor de Oracle remoto, la tabla EMPLOY se crea mediante tipos de datos de Oracle. La tabla siguiente muestra las correlaciones de los tipos de datos de DB2 con los tipos de datos de Oracle para las columnas especificadas en el ejemplo.

Tabla 7. Un ejemplo de correlaciones de tipos de datos inversas de la base federada con Oracle

Columna	Tipo de datos de DB2 especificado en la sentencia CREATE TABLE	Tipo de datos de Oracle utilizados en la tabla remota
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)

Modificación de tablas remotas utilizando DDL transparente

Puede modificar tablas remotas de orígenes de datos que se crearon a través de la base de datos federada utilizando DDL transparente. No puede modificar las tablas que se crearon directamente en el origen de datos remoto.

Antes de empezar

El ID de autorización de las sentencias de DDL transparente debe incluir al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Autoridad CREATETAB en la base de datos y privilegio USE en el espacio de tabla, así como uno de los privilegios siguientes:
 - Autoridad IMPLICIT_SCHEMA en la base de datos, si el nombre de esquema implícito o explícito de la tabla no existe
 - Privilegio CREATEIN en el esquema, si el nombre de esquema de la tabla hace referencia a un esquema existente

Para emitir sentencias de DDL transparente, su ID de autorización debe contener los privilegios necesarios en el apodo (para que el servidor federado acepte la petición), y los privilegios comparables en el origen de datos remoto (para que el origen de datos acepte la petición).

Restricciones

Las restricciones siguientes son aplicables a la modificación de una tabla remota mediante DDL transparente:

- No puede modificar las tablas que se crearon originalmente en el origen de datos remoto.
- No puede modificar o descartar una clave primaria existente en una tabla remota.
- Al alterar una tabla remota se invalida cualquier paquete dependiente del apodo asociado a la tabla remota.
- El origen de datos remoto debe dar soporte a los cambios en la sentencia ALTER TABLE. Por ejemplo, suponga que el origen de datos remoto no da soporte a las claves primarias. Dependiendo de la forma en que el origen de datos responda a las peticiones a las que no da soporte, puede que se devuelva un error o puede que la petición sea ignorada.
- No puede especificar un comentario en una columna.
- No puede generar contenidos de columna.
- Puede especificar una clave primaria, pero no podrá especificar restricciones de clave foránea o de comprobación. Las columnas utilizadas para la clave primaria deben ser NOT NULL, y no pueden incluir columnas que contengan objetos LOB.
- No puede modificar los parámetros de columnas existentes como, por ejemplo, el tipo o longitud de los datos.
- La cláusula DEFAULT de la sentencia ALTER TABLE no recibe soporte.

Acerca de esta tarea

Puede utilizar el Centro de control de DB2 o la sentencia ALTER TABLE para modificar las tablas creadas mediante IBM InfoSphere Federation Server utilizando

DDL transparente. Utilice el Centro de control de DB2 para evitar la especificación de un parámetro u opción que no reciba soporte. Utilizando la sentencia ALTER TABLE podrá:

- Añadir nuevas columnas
- Añadir la clave primaria de la tabla

No utilice la sentencia ALTER TABLE para añadir o modificar opciones de columna. En su lugar, utilice la sentencia ALTER NICKNAME.

Procedimiento

Para modificar una tabla remota utilizando DDL transparente, emita la sentencia ALTER TABLE:

Ejemplo: Desea añadir una clave primaria en una tabla remota EMPLOYEE que creó utilizando DDL transparente. Utilice la sentencia ALTER TABLE siguiente para modificar la tabla:

```
ALTER TABLE EMPLOYEE
  ADD PRIMARY KEY (EMP_NO, WORK_DEPT)
```

Las columnas utilizadas para la clave primaria deben ser NOT NULL, y no pueden ser columnas que contengan objetos LOB.

Ejemplo: Desea añadir las columnas ORDER_DATE y SHIP_DATE a la tabla remota SPALTEN que creó utilizando DDL transparente. Utilice la sentencia ALTER TABLE siguiente para crear la tabla:

```
ALTER TABLE SPALTEN
  ADD COLUMN ORDER_DATE DATE
  ADD COLUMN SHIP_DATE DATE
```

Descarte de tablas remotas utilizando DDL transparente

Puede descartar tablas remotas de orígenes de datos que se crearon a través de la base de datos federada utilizando DDL transparente. No puede descartar las tablas que se crearon directamente en el origen de datos remoto.

Antes de empezar

El ID de autorización de las sentencias de DDL transparente debe incluir al menos uno de los privilegios siguientes:

- Autoridad SYSADM o DBADM
- Autoridad CREATETAB en la base de datos y privilegio USE en el espacio de tabla, así como uno de los privilegios siguientes:
 - Autoridad IMPLICIT_SCHEMA en la base de datos, si el nombre de esquema implícito o explícito de la tabla no existe
 - Privilegio CREATEIN en el esquema, si el nombre de esquema de la tabla hace referencia a un esquema existente

Para emitir sentencias de DDL transparente, su ID de autorización debe contener los privilegios necesarios en el apodo (para que el servidor federado acepte la petición), y los privilegios comparables en el origen de datos remoto (para que el origen de datos acepte la petición).

Restricciones

No puede descartar las tablas que se crearon originalmente en el origen de datos remoto.

Acerca de esta tarea

Para descartar tablas remotas que fueron creadas a través de la base de datos federada utilizando DDL transparente, puede utilizar el Centro de control de DB2 o la sentencia DROP TABLE.

Al descartar un apodo para una tabla remota creada utilizando DDL transparente se descarta simplemente el apodo local para dicha tabla. La sentencia DROP NICKNAME no descarta la tabla remota. Se debe utilizar la sentencia DROP TABLE para descartar la tabla remota.

Al descartar una tabla remota, primero se suprime la tabla en el origen de datos y después se suprime el apodo correspondiente para la tabla remota en la base de datos federada. Al suprimir el apodo se invalida cualquier paquete basado en dicho apodo.

Procedimiento

Para descartar una tabla remota, emita la sentencia DROP TABLE.

Ejemplo: Para descartar una tabla denominada SPALTEN, emita la sentencia DROP siguiente:

```
DROP TABLE SPALTEN
```

dónde *SPALTEN* es el nombre local para la tabla remota.

Capítulo 8. Gestión de transacciones en un sistema federado

El proceso de transacciones de un sistema federado permite leer y actualizar las bases de datos en una sola transacción, a la vez que se mantiene la coherencia de los datos. Los sistemas federados son compatibles con protocolos de confirmación en una fase y en dos fases. Al administrar el sistema federado, debe configurar el protocolo apropiado.

Descripción del soporte de transacción de un sistema federado

Es conveniente que conozca los conceptos sobre el proceso de transacciones de un entorno distribuido de DB2 Database para Linux, UNIX y Windows para comprender las transacciones de un sistema federado.

Para comprender el proceso de transacciones de un sistema federado, es conveniente que conozca los conceptos siguientes sobre el proceso de transacciones distribuidas:

- Unidad de trabajo (UOW)
- Unidad de trabajo remota (RUOW)
- Unidad de trabajo distribuida (DUOW)
- Actualización múltiple
- Gestor de transacciones (TM)
- Gestor de recursos (RM)
- Conexión de tipo 1
- Conexión de tipo 2
- Confirmación en una fase
- Confirmación en dos fases

Estos conceptos son los mismos en los sistemas de base de datos federados y no federados. Sin embargo, el ámbito de aplicación de cada concepto cambia en un sistema federado.

Por ejemplo, una unidad de trabajo comienza implícitamente cuando se lee o escribe cualquier dato en una base de datos. Para una unidad de trabajo de un sistema federado, la base de datos puede ser una base de datos federada o una base de datos de origen de datos. Para una unidad de trabajo distribuida de un sistema federado, el usuario puede acceder a una base de datos federada o a una base de datos de origen de datos.

La aplicación debe finalizar una unidad de trabajo emitiendo una sentencia COMMIT o ROLLBACK, sin importar el número de bases de datos a las que se accede. La sentencia COMMIT convierte en permanentes todos los cambios realizados dentro de una unidad de trabajo. La sentencia ROLLBACK elimina esos cambios en una base de datos. Los cambios hechos por una unidad de trabajo pasan a ser visibles para las demás aplicaciones después de una operación de confirmación satisfactoria.

Recomendación: realice siempre una confirmación o retrotracción explícita de las unidades de trabajo en las aplicaciones.

En una unidad de trabajo distribuida en la que se produzcan actualizaciones en varias bases de datos situadas en ubicaciones diferentes, los datos deben ser coherentes. Normalmente se utiliza la actualización múltiple o el protocolo de confirmación en dos fases para asegurar la coherencia de los datos entre diversas bases de datos dentro de una unidad de trabajo distribuida.

Las transacciones federadas son compatibles con el protocolo de confirmación de una fase y el protocolo de confirmación en dos fases. La opción de servidor DB2_TWO_PHASE_COMMIT permite utilizar la confirmación en dos fases para los orígenes de datos siguientes:

- Orígenes de datos de la familia de productos DB2
- Informix
- Oracle
- Sybase
- MS SQL Server

Cuando un origen de datos está declarada como origen de datos federado con confirmación en dos fases, es decir, el valor de la opción DB2_TWO_PHASE_COMMIT es "S", las operaciones de confirmación realizadas para este origen de datos utilizan el protocolo de confirmación en dos fases, aunque sea una transacción de actualización simple o de actualización múltiple.

Cuando un origen de datos está declarado como origen de datos federado con confirmación de una fase (el valor por omisión), y se realiza una transacción de actualización simple, las operaciones de confirmación realizadas para este origen de datos utilizan el protocolo de confirmación de una fase.

En el ejemplo siguiente de operación de confirmación de una fase, Oracle está definido como origen de datos con confirmación de una fase:

```
SELECT * FROM apodo_oracle  
UPDATE apodo_oracle  
COMMIT
```

En el ejemplo siguiente de operación de confirmación en dos fases, Oracle y DRDA están definidos como orígenes de datos con confirmación en dos fases:

```
SELECT * FROM apodo_oracle  
UPDATE apodo_oracle
```

```
SELECT * FROM apodo_drda  
UPDATE apodo_drda  
COMMIT
```

Qué es una actualización en un sistema federado

En un sistema federado, una actualización no es simplemente una transacción que incluye una sentencia INSERT, UPDATE o DELETE. Existen determinadas operaciones que se consideran actualizaciones en un sistema federado y determinados tipos de actualizaciones que están permitidos en un sistema federado.

En un sistema federado, las actualizaciones se pueden realizar localmente o de forma remota.

- Las actualizaciones locales son las que se realizan sobre tablas o vistas de DB2 que no hacen referencia a apodos

- Las actualizaciones remotas son las que se realizan sobre objetos situados en un origen de datos remoto. Son orígenes de datos remotos:
 - Otra base de datos o instancia de DB2 Database para Linux, UNIX y Windows situada en el servidor federado
 - Otra base de datos o instancia de DB2 Database para Linux, UNIX y Windows situada en otro servidor
 - Orígenes de datos que no sean DB2 Database para Linux, UNIX y Windows, tales como DB2 para System i, Informix, Oracle y Teradata

Existen cuatro tipos de acciones que son tratadas como transacciones de actualización por el servidor federado. La tabla siguiente muestra las actualizaciones que puede realizar en un sistema federado.

Tabla 8. Tipos de actualizaciones y emplazamiento donde se realizan las actualizaciones

Tipo de acción	Emplazamiento		Explicación
	local	remoto	
Actualización local (DDL y DML)	S	N	Actualización de un objeto en la base de datos federada.
Actualización remota (apodo)	N	S	Actualización en un objeto de un origen de datos remoto para el cual se ha creado un apodo.
SQL dinámico en sesiones de paso a través	N	S	Actualización en un objeto de un origen de datos remoto. No puede utilizar una sesión de paso a través para actualizar objetos locales. Incluso las consultas SELECT enviadas en sesiones de paso a través se consideran que son una acción de actualización.
DDL transparente	S	S	Par de transacciones por el que se crean, alteran o eliminan tablas remotas y sus correspondientes apodos en una base de datos federada. Por ejemplo, un par de transacciones por el que se crean una tabla remota en un origen de datos y un apodo en el servidor federado.

Qué es una transacción de actualización en una sesión de paso a través

Un servidor federado trata como actualizaciones todas las sentencias de SQL dinámico emitidas en una sesión de paso a través. Este comportamiento asegura la integridad de los datos.

Si una sentencia de SQL dinámico emitida en una sesión de paso a través se ejecuta satisfactoriamente, la transacción se registra como actualización. El SQL puede ser cualquier tipo de sentencia, incluidas las sentencias SELECT.

Orígenes de datos con confirmación automática de las sentencias de DDL

Algunos orígenes de datos, tales como Oracle, confirman automáticamente la transacción actual en el emplazamiento del origen de datos como parte de la ejecución de una sentencia de DDL.

Si crea una tabla remota utilizando DDL transparente o en una sesión de paso a través, estos orígenes de datos no pueden retrotraer la tabla remota una vez creada la tabla. Debe suprimir manualmente la tabla remota.

Funciones definidas por el usuario que son enviadas al origen de datos para su proceso

Si una función remota definida por usuario realiza una actualización en un origen de datos, el servidor federado no detecta la actualización.

Debido a que el servidor federado no trata estas funciones definidas por el usuario como sentencias de actualización, toda la protección a nivel de sentencia que aplica el sistema federado a las operaciones de actualización no es aplicable. En consecuencia, la integridad de los datos puede estar en peligro en algunas situaciones.

Importante: no se puede asegurar la integridad de los datos cuando una función definida por el usuario emite una actualización para un origen de datos.

Capítulo 9. Realización de transacciones de confirmación en dos fases

Puede utilizar la capacidad de confirmación en dos fases de un sistema federado para actualizar datos de uno o varios orígenes de datos en una única transacción.

Confirmación en dos fases para transacciones federadas

Un sistema federado puede utilizar la confirmación en dos fases para transacciones que acceden a uno o varios orígenes de datos. La confirmación en dos fases utiliza el protocolo estándar XA de X/Open para coordinar el proceso de transacciones de unidad de trabajo distribuida.

En una operación de confirmación en dos fases, el proceso de confirmación se produce en dos fases: la fase de preparación y la fase de confirmación. Durante la fase de preparación de un sistema federado, un servidor federado sondea todos los orígenes de datos de confirmación federada en dos fases que intervienen en una transacción. Esta actividad de sondeo verifica si cada origen de datos está preparada para confirmar o retrotraer los datos. Durante la fase de confirmación, el servidor federado ordena a cada origen de datos de confirmación en dos fases que confirme los datos o retrotraiga la transacción.

En la confirmación en una fase, se actualizan por separado varios orígenes de datos utilizando una operación de confirmación diferente. Esto puede producir problemas de sincronización de datos si algunos orígenes de datos se actualizan satisfactoriamente y otras no lo hacen.

Por ejemplo, si una transacción retira fondos de una cuenta y los ingresa en otra cuenta mediante una confirmación en una fase, el sistema puede confirmar satisfactoriamente la operación de retirada de fondos y no lograr confirmar la operación de ingreso. La operación de ingreso se puede retrotraer, pero no así la operación de retirada de fondos, pues ya se ha confirmado satisfactoriamente. El resultado es una "pérdida" virtual de fondos.

En una confirmación en dos fases, las transacciones de retirada e ingreso se preparan juntas, y se confirman o retrotraen juntas. Como consecuencia, la integridad de los importes de los fondos permanece inalterada.

Planificación para la confirmación federada en dos fases

La confirmación federada en dos pasos no proporciona ventajas en todos los entornos de trabajo. Además, es necesario tener en cuenta varios factores antes de decidir desplegar la confirmación federada en dos pasos.

Para utilizar la confirmación en dos fases para transacciones federadas, debe tener en cuenta las cuestiones siguientes:

- Si el sistema operativo y los orígenes de datos que utiliza son compatibles con la confirmación en dos fases para transacciones federadas.
- Si su entorno de trabajo necesita la confirmación en dos fases para transacciones federadas.

Para decidir si la confirmación en dos fases es apropiada para su entorno de trabajo, es necesario que conozca cómo trabaja la confirmación en dos fases para las transacciones federadas y los problemas que ese mecanismo soluciona.

- Cómo configurar el servidor federado y los orígenes de datos compatibles para utilizar la confirmación en dos fases para las transacciones federadas.

Existen unos requisitos básicos respecto al servidor federado y los orígenes de datos para utilizar la confirmación en dos fases para transacciones federadas, así como consideraciones sobre el rendimiento que debe tener en cuenta al desplegar la confirmación en dos fases.

- Para resolver manualmente las transacciones dudosas, es necesario que conozca el funcionamiento interno de la confirmación en dos fases para transacciones federadas.

La confirmación en dos fases para transacciones federadas puede resolver problemas sin la intervención del usuario. Pero cuando existen periodos prolongados de falta de disponibilidad de la red, errores de hardware o una necesidad urgente de liberar recursos del sistema, puede resolver manualmente problemas mediante el *proceso heurístico*.

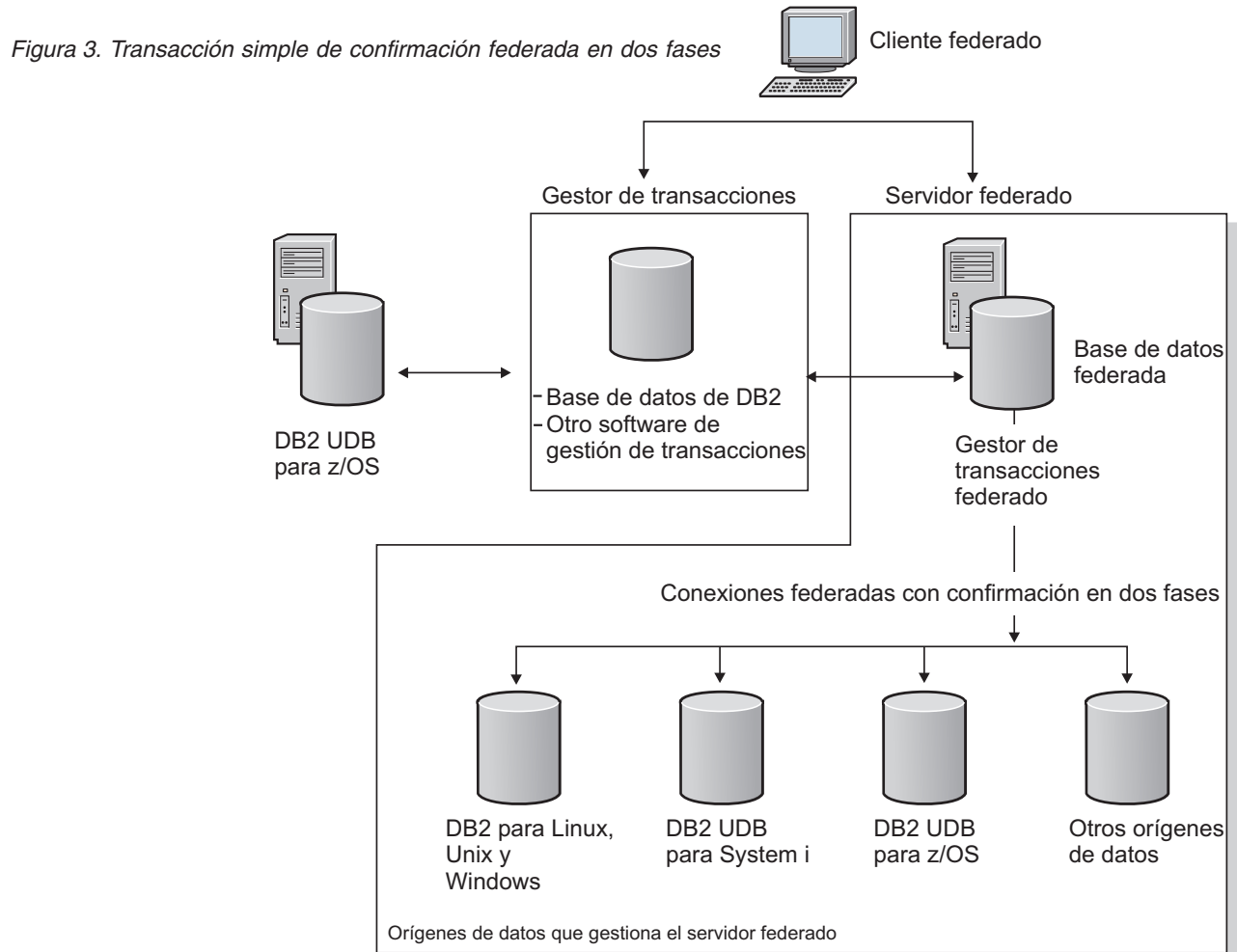
Arquitectura federada para la confirmación en dos fases

La confirmación federada en dos fases está basada en la confirmación en dos fases existente en DB2. En la confirmación en dos fases, el modelo Distributed Transaction Processing (DTP) de X/Open tiene varios componentes: identificadores de transacciones, gestores de transacciones y gestores de recursos. En los sistemas federados con confirmación en dos fases existe un componente adicional: el gestor de transacciones federado.

Un servidor federado pasa a ser un gestor de transacciones federado cuando el servidor coordina la actividad de una o más orígenes de datos remotos que hacen uso del protocolo de confirmación en dos fases. Un gestor de transacciones federado realiza algunas funciones de gestión de transacciones a petición del gestor de transacciones. El cliente o aplicación por el que se inicia una transacción de unidad de trabajo distribuida y el gestor de transacciones no conocen la actividad que el gestor de transacciones federado coordina en los orígenes de datos remotos. El gestor de transacciones federado se comunica con gestores de transacciones de bases de datos DB2 mediante una interfaz XA. Además de los requisitos de X/Open para la confirmación en dos fases, la base de datos del gestor de transacciones debe ser accesible desde la instancia federada. Los gestores de recursos siguen las instrucciones proporcionadas por un gestor de transacciones federado para confirmar o retrotraer una transacción.

La figura siguiente muestra un ejemplo de una transacción simple de confirmación en dos fases realizada en un sistema federado típico, desde la iniciación del cliente

hasta las actualizaciones de orígenes de datos.



En la figura anterior, la conexión del cliente con el gestor de transacciones es una conexión de tipo 2. Cada conexión de base de datos tiene también su propio valor de punto de sincronismo. Un punto de sincronismo es un punto en el tiempo en el que todos los datos recuperables a los que accede un programa son coherentes. Las conexiones en dos fases con punto de sincronismo pueden trabajar con transacciones de unidad de trabajo distribuida con actualizaciones en varios orígenes de datos.

Cuando el cliente se conecta a la base de datos DB2, el gestor de transacciones tiene conocimiento de la transacción, pero no es necesaria ninguna coordinación adicional por parte del servidor federado. Cuando el servidor federado se conecta a los orígenes de datos utilizando el protocolo de confirmación en dos fases, el servidor federado se convierte en el gestor de transacciones federado. El servidor federado supervisa y coordina las confirmaciones en dos fases. En este momento, el gestor de transacciones no tiene conocimiento de las transacciones de confirmación en dos fases realizadas con los orígenes de datos. El gestor de transacciones solamente sabe que se está procesando una transacción individual con el servidor federado.

Los orígenes de datos no son capaces de iniciar una resincronización si se produce un error en un sistema federado. El servidor federado inicia el proceso de resincronización.

Los resultados pueden ser imprevisibles si intenta acceder a un origen de datos utilizando varias vías de acceso en la misma transacción durante una confirmación federada en dos fases. Por ejemplo, si el servidor federado es un gestor de recursos respecto de un gestor de transacciones externo, se puede acceder al origen de datos indirectamente desde el servidor federado y directamente como gestor de recursos del gestor de transacciones. En este caso, el origen de datos puede no ser capaz de distinguir si estas dos vías de acceso pertenecen a la misma transacción global. El origen de datos podría crear dos entradas de transacción para la misma transacción global y tratar las transacciones como separadas entre sí, lo que posiblemente produciría resultados imprevisibles. El origen de datos podría también detectar que las dos vías de acceso pertenecen a la misma transacción global, y rechazar la segunda vía.

Confirmación en dos fases para transacciones federadas - ejemplos

Un sistema federado que hace uso de la confirmación en dos fases se puede configurar de varias maneras diferentes. La configuración elegida depende de la solución necesaria.

Las configuraciones pueden utilizar conexiones de Tipo 1 o de Tipo 2.

Las conexiones de Tipo 1 son conexiones en las que un proceso de aplicación está conectado a un servidor de aplicaciones de acuerdo con las normas de la unidad de trabajo remota.

Las conexiones de Tipo 2 son conexiones en las que un proceso de aplicación está conectado a un servidor de aplicaciones y establece las normas para la unidad de trabajo distribuida dirigida por la aplicación. El servidor de aplicaciones se convierte entonces en el servidor actual del proceso.

La figura siguiente muestra una conexión DB2 de Tipo 1 con un servidor federado que actúa como gestor de transacciones.

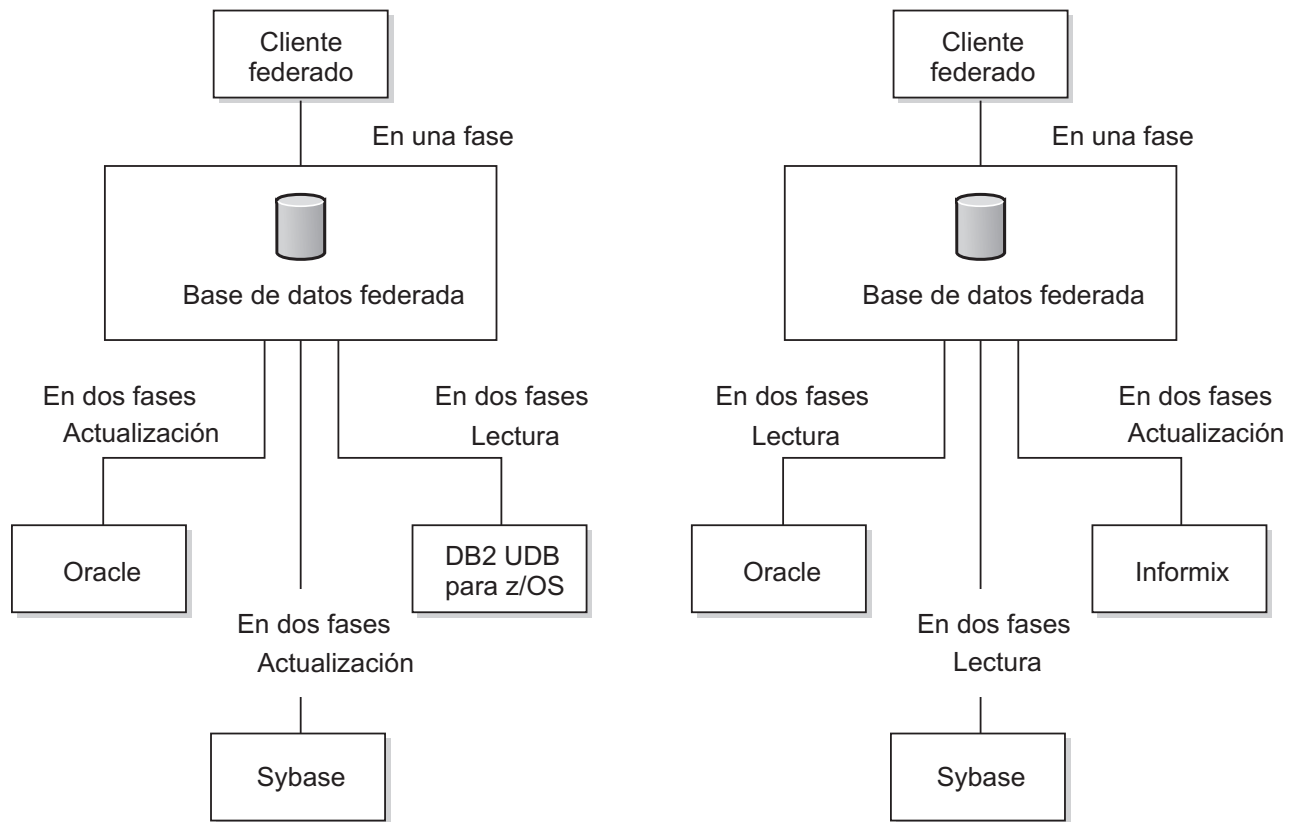


Figura 4. Conexión DB2 de Tipo 1 con un servidor federado que actúa como gestor de transacciones

La figura siguiente muestra una conexión DB2 de Tipo 2 con un servidor federado que actúa como gestor de recursos. En esta configuración, todos los orígenes de datos federados deben ser compatibles con la confirmación federada en dos fases y estar habilitados para ella.

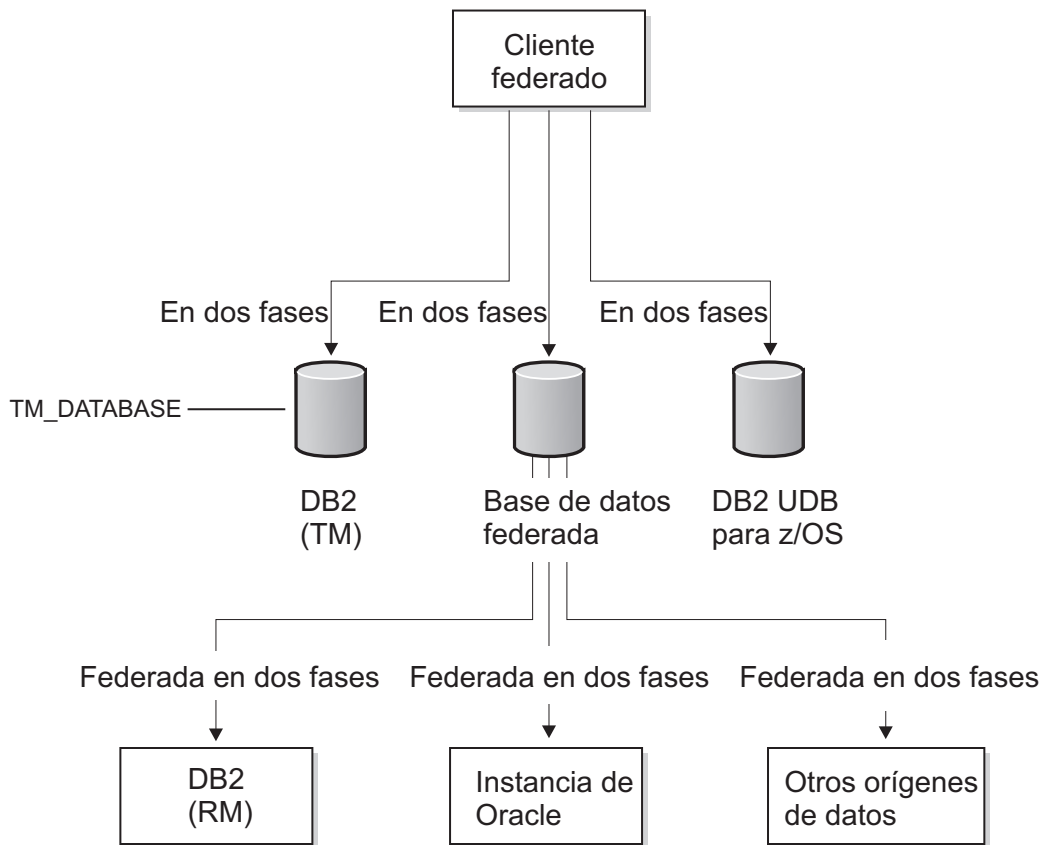


Figura 5. Conexión DB2 de Tipo 2 con un servidor federado que actúa como gestor de recursos

La figura siguiente muestra una conexión DB2 de Tipo 2 con un servidor federado que actúa como gestor de transacciones.

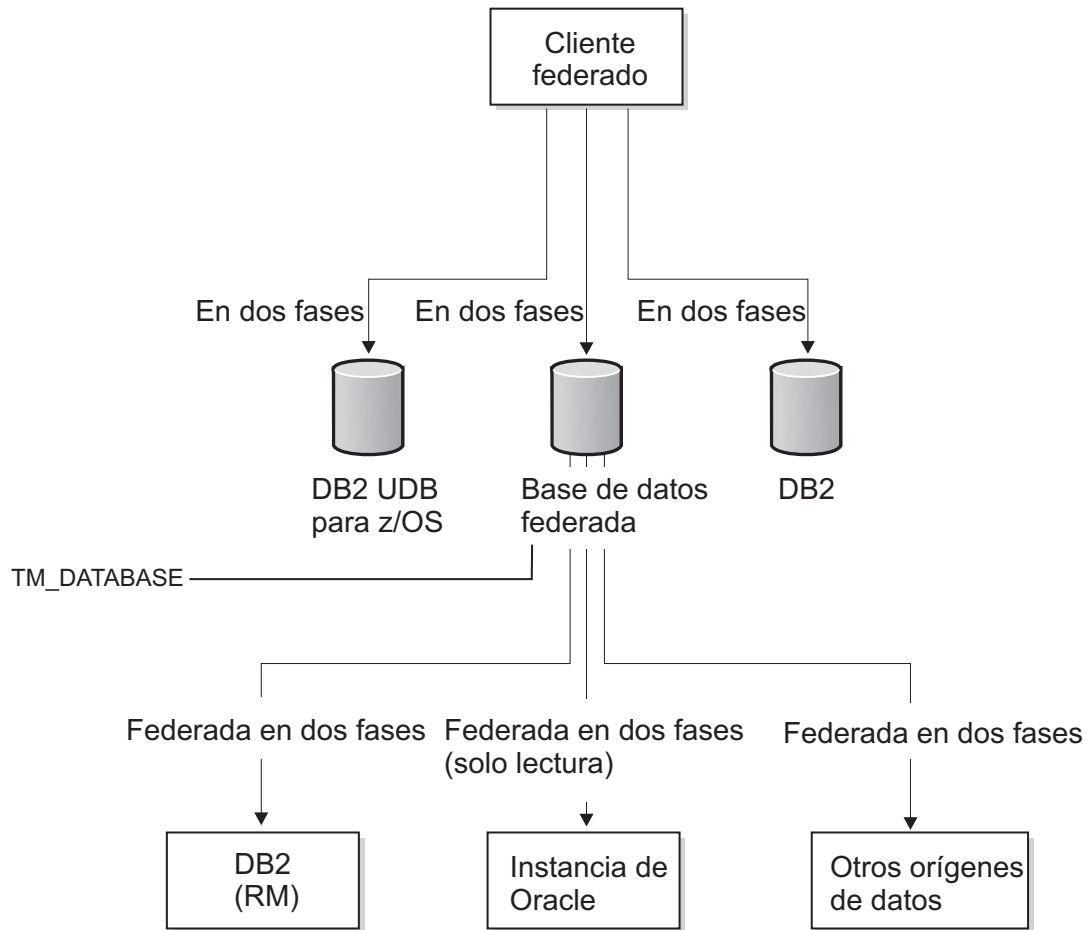


Figura 6. Conexión DB2 de Tipo 2 con un servidor federado que actúa como gestor de transacciones

La figura siguiente muestra una conexión XA con un servidor federado que actúa como gestor de recursos.

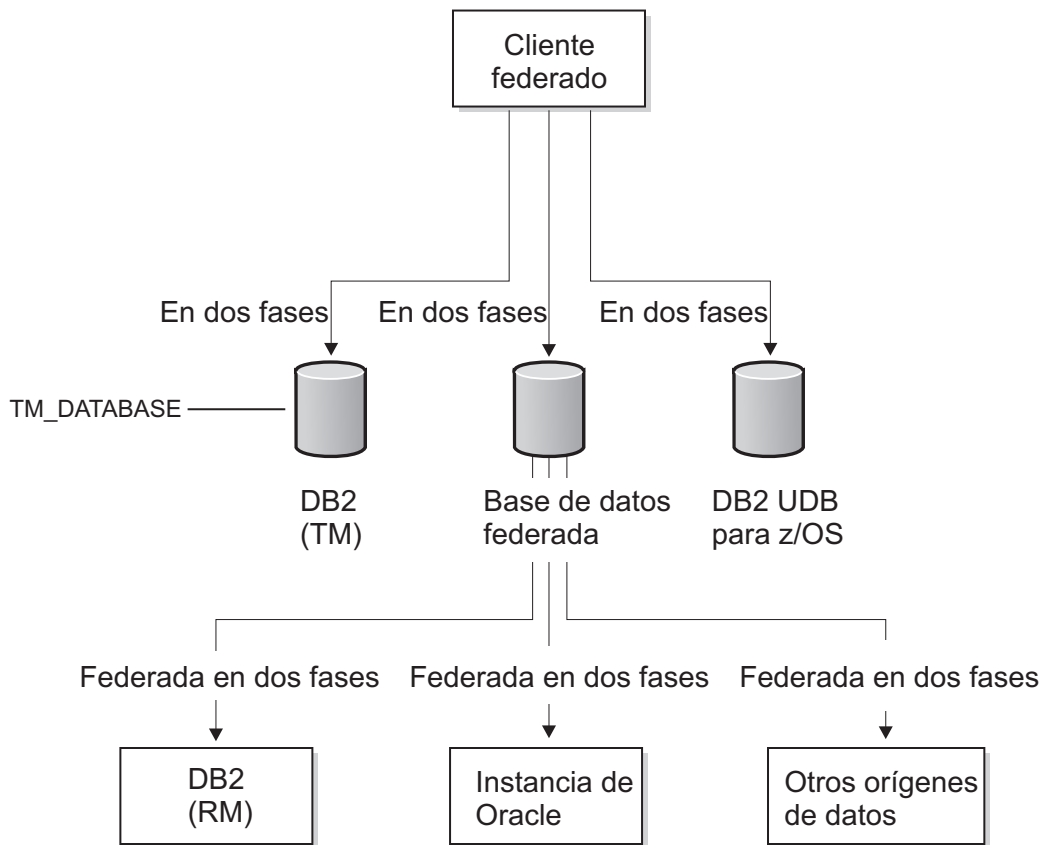


Figura 7. Conexión XA con un servidor federado que actúa como gestor de recursos

Proceso de las transacciones federadas con confirmación en dos fases

El servidor federado mantiene la coherencia de los datos y la atomicidad de los orígenes de datos que gestiona. El rango de transacciones posibles depende del tipo de conexión y de si el servidor federado es el gestor de transacciones o el gestor de recursos de la conexión.

La atomicidad es un principio de la base de datos por el cual se definen grupos de operaciones dentro de transacciones indivisibles. Esta característica asegura en todo momento la coherencia de la base de datos, pues si falla una operación individual dentro de la transacción indivisible, fallará la transacción completa en lugar de comprometer la integridad de los datos debido a un cambio parcial.

Por ejemplo, una transacción para transferir fondos desde una cuenta a otra comprende la retirada de fondos de la primera cuenta y la adición de fondos a la segunda cuenta. Solamente si la retirada de fondos se realiza satisfactoriamente, los fondos esencialmente dejan de existir en la primera cuenta.

El servidor federado procesa las peticiones de actualización federada de acuerdo con reglas estrictas. Una actualización federada es una de las acciones siguientes:

- Una operación federada de inserción, actualización o supresión en la que el correspondiente origen de datos es compatible con operaciones de inserción, actualización o supresión. Por ejemplo, algunos orígenes de datos no admiten

operaciones de actualización. Algunos orígenes de datos son de sólo lectura, en cuyo caso no están permitidas las operaciones federadas de inserción, actualización o supresión.

- Una operación exitosa de paso a través realizada dentro de una sesión de paso a través.
- Una operación de DDL transparente, que se considera que es al mismo tiempo una actualización local y una actualización federada porque realiza una actualización de la base de datos de forma local y remota.
- Un procedimiento almacenado federado con MODIFY SQL ACCESS.

Nota: No utilice varias vías de acceso para acceder al mismo origen de datos dentro de una transacción individual. Una transacción de esta clase podría llegar a una situación de punto muerto. Es decir, la transacción puede quedar bloqueada. Por ejemplo, no utilice varios servidores federados para hacer referencia a los mismos orígenes de datos en la misma transacción.

La tabla siguiente describe qué ocurre en una transacción individual de unidad de trabajo distribuida, incluido el tipo de conexión, el tipo de confirmación, el rol del servidor federado en la transacción, qué operaciones están permitidas y cómo se puede utilizar el DDL transparente.

Tabla 9. Qué ocurre en una transacción individual de unidad de trabajo distribuida

Tipo de confirmación	Tipo de conexión	Rol del servidor federado	Operaciones	DDL transparente
Una fase	Transacción local DB2 de Tipo 1 o XA	Gestor de subtransacciones. También actúa como coordinador de transacciones, determina el resultado de la transacción y lo entrega a cada gestor de recursos participante.	Están permitidas las operaciones de lectura con confirmación en una fase y confirmación en dos fases. Un origen de datos con confirmación en una fase se puede actualizar siempre que sea la única actualización de la transacción.	Se permite y gestiona de acuerdo con las reglas para orígenes de datos con confirmación en una fase. Cada sentencia emitida debe ser la única actualización existente en la transacción con confirmación en una fase. La actualización no puede coexistir con otras actualizaciones de orígenes de datos con confirmación en dos fases existentes en la misma transacción. Es muy recomendable emitir sentencias COMMIT o ROLLBACK antes y después de producirse transacciones de DDL transparente.

Tabla 9. Qué ocurre en una transacción individual de unidad de trabajo distribuida (continuación)

Tipo de confirmación	Tipo de conexión	Rol del servidor federado	Operaciones	DDL transparente
Dos fases	Transacción local DB2 de Tipo 1 o XA	Gestor de transacciones. También actúa como coordinador de transacciones, determina el resultado de la transacción y lo entrega a cada gestor de recursos participante.	Están permitidas las operaciones de lectura con confirmación en una fase y confirmación en dos fases. Se pueden actualizar varios orígenes de datos con confirmación en dos fases.	Se permite y gestiona de acuerdo con las reglas para orígenes de datos con confirmación en dos fases. La actualización puede coexistir con otras actualizaciones de orígenes de datos con confirmación en dos fases o una fase existentes en la misma transacción.
Una fase	Transacción global DB2 de Tipo 2 o XA	Puede actuar como gestor de transacciones. Si no es el gestor de transacciones, solamente transfiere el resultado del coordinador de transacciones externo a cada gestor de recursos participante.	Están permitidas las operaciones de lectura con confirmación en una fase y confirmación en dos fases. Las actualizaciones en una fase no están permitidas excepto para transacciones coordinadas por DB2 que pueden realizar actualizaciones federadas en una fase a través de una conexión entrante de dos fases de Distributed Relational Database Architecture (DRDA).	Se permite y gestiona de acuerdo con las reglas para orígenes de datos con confirmación en una fase. Cada sentencia emitida debe ser la única actualización existente en la transacción con confirmación en una fase. La actualización no puede coexistir con otras actualizaciones de orígenes de datos con confirmación en dos fases existentes en la misma transacción. Es muy recomendable emitir sentencias COMMIT o ROLLBACK antes y después de producirse transacciones de DDL transparente.

Tabla 9. Qué ocurre en una transacción individual de unidad de trabajo distribuida (continuación)

Tipo de confirmación	Tipo de conexión	Rol del servidor federado	Operaciones	DDL transparente
Dos fases	Transacción global DB2 de Tipo 2 o XA	Puede actuar como gestor de transacciones. Si no es el gestor de transacciones, solamente transfiere el resultado del coordinador de transacciones externo a cada gestor de recursos participante.	Están permitidas las operaciones de lectura con confirmación en una fase y confirmación en dos fases. Se pueden actualizar varios orígenes de datos con confirmación en dos fases.	Se permite y gestiona de acuerdo con las reglas para orígenes de datos con confirmación en dos fases. La actualización puede coexistir con otras actualizaciones de orígenes de datos con confirmación en dos fases o una fase existentes en la misma transacción.

Mantenimiento de la coherencia de los datos y de la atomicidad

Los servidores federados intentan asegurar la coherencia de los datos y mantener la atomicidad de las transacciones en los orígenes de datos.

Se produce un error (SQL30090, código de razón 18) cuando surge un conflicto entre el punto de sincronización de una aplicación y la posibilidad de actualización de un origen de datos.

Las actualizaciones locales que incluyen DDL realizadas en la base de datos federada no se pueden mezclar dentro de la misma transacción con una actualización hecha en un origen de datos federado con confirmación en una fase. DDL transparente

Utilización de DDL y DDL transparente

Las actualizaciones locales que incluyen DDL realizadas en la base de datos federada no se pueden mezclar dentro de la misma transacción con una actualización hecha en un origen de datos federado con confirmación en una fase. El DDL transparente es una excepción. En el DDL transparente, están permitidas las actualizaciones locales y las actualizaciones de origen de datos sin importar el tipo de conexión y si el origen de datos está configurado para la confirmación en una fase o en dos fases.

El DDL transparente crea una tabla en un origen de datos remoto y un apodo para la tabla remota en la base de datos federada local. Un servidor federado trata las transacciones de DDL transparente como actualizaciones.

El DDL transparente proporciona la capacidad para crear y modificar tablas remotas mediante el sistema de base de datos DB2, sin necesidad de utilizar sesiones de paso a través. Las sentencias de SQL para el DDL transparente son CREATE TABLE, ALTER TABLE y DROP TABLE. Por ejemplo, una sentencia CREATE TABLE de DDL transparente crea una tabla remota en el origen de datos y un apodo para esa tabla en el servidor federado. La sentencia contiene una operación de actualización local y una operación de actualización remota.

Algunos orígenes de datos, tales como Oracle, no permiten DDL transparente en una conexión federada con confirmación en dos fases.

Habilitación de la confirmación en dos fases para las transacciones federadas

Para utilizar la confirmación federada en dos fases para determinados orígenes de datos, debe habilitar los servidores federados correspondientes. El proceso de habilitación comprende preparar el servidor federado y modificar la definición del servidor en el origen de datos.

Antes de empezar

- Cuando habilita la confirmación federada en dos fases para un origen de datos, aumenta el número de registros que se escriben en el archivo de registro de la base de datos del servidor federado y en el archivo de registro de la base de datos del origen de datos. Tenga en cuenta el efecto que esto tiene en la administración y mantenimiento de estos archivos de registro para asegurarse de que estos archivos cumplen las políticas locales.
- El origen de datos que desee utilizar debe ser compatible con la confirmación federada en dos fases.
- La confirmación en dos fases federada no se admite en el de procesos paralelos masivos (MPP) ni en el entorno con limitaciones, con la excepción de Sybase. Sólo en el caso de Sybase en UNIX, la confirmación en dos fases federada se admite en el entorno con limitaciones.
- Para orígenes de datos DB2 para System i, versión 5.3 y versiones anteriores, y DB2 para z/OS, asegúrese de que el parámetro de configuración SPM_NAME está establecido en el valor por omisión, que es el nombre de host del servidor. El valor por omisión de SPM_NAME es una variante de los primeros siete caracteres del nombre del host TCP/IP. En DB2 para System i, versión 5.4 y versiones posteriores no es necesario establecer SPM_NAME.

Acerca de esta tarea

La opción de servidor DB2_TWO_PHASE_COMMIT permite utilizar la confirmación en dos fases para orígenes de datos. Utilice la sentencia CREATE SERVER para registrar una definición de servidor de un origen de datos. El valor definido para DB2_TWO_PHASE_COMMIT se conserva para todas las conexiones que se establezcan utilizando esa definición de servidor. Puede cambiar el valor en cualquier momento utilizando la sentencia ALTER SERVER. Una vez ejecutada satisfactoriamente la sentencia CREATE SERVER o ALTER SERVER, el nuevo valor se puede utilizar en peticiones subsiguientes de conexión de salida.

Los clientes y programas de aplicación pueden utilizar la sentencia SET SERVER OPTION para modificar temporalmente el valor de la opción de servidor DB2_TWO_PHASE_COMMIT. La sentencia SET SERVER OPTION se debe ejecutar inmediatamente después de establecer conexión con la base de datos del servidor federado y antes de establecer cualquier conexión con orígenes de datos remotos. El mandato solamente es efectivo mientras está activa la conexión con la base de datos federada. No puede cambiar la opción de servidor DB2_TWO_PHASE_COMMIT una vez que el servidor federado ha establecido conexión con el origen de datos remoto.

Cuando incluye la opción `XA_OPEN_STRING_OPTIONS` en una sentencia `CREATE SERVER`, puede intercalar información especializada en la cadena `XA_OPEN` por omisión. Esta información intercalada puede ser cualquiera de las clases siguientes:

- Identificadores exclusivos para las transacciones, además de los proporcionados por IBM InfoSphere Federation Server.
- Parámetros definidos por el usuario para el manejo de transacciones
- Una cadena de caracteres definida por el usuario para añadir a la petición `XA_OPEN`

Cuando se realiza una llamada `XA_OPEN`, usualmente al comienzo de la primera transacción con un origen de datos remoto donde se utiliza la confirmación en dos fases, el derivador añade el valor de la cadena de caracteres definida por el usuario a la cadena `XA_OPEN` por omisión para la llamada `XA_OPEN`.

Puede incluir tanto `DB2_TWO_PHASE_COMMIT` como `XA_OPEN_STRING_OPTIONS` en una sentencia `CREATE SERVER`, `SET SERVER` o `ALTER SERVER`.

Procedimiento

En general, para habilitar la confirmación en dos fases para un origen de datos federado, debe seguir estos pasos:

1. Ejecute la sentencia `CREATE SERVER`, `ALTER SERVER` o `SET SERVER` con la opción `DB2_TWO_PHASE_COMMIT` establecida en S.
2. Opcional: Ejecute la sentencia `CREATE SERVER`, `ALTER SERVER` o `SET SERVER` con la opción `XA_OPEN_STRING_OPTIONS`.

Ejemplos de la opción de servidor

Este ejemplo muestra cómo habilitar la confirmación en dos fases mediante la sentencia `CREATE SERVER`:

```
CREATE SERVER Net8_Server TYPE ORACLE VERSION 8.1.7 WRAPPER NET8
OPTIONS (DB2_TWO_PHASE_COMMIT 'S');
```

Este ejemplo muestra cómo inhabilitar la confirmación en dos fases mediante la sentencia `ALTER SERVER`:

```
ALTER SERVER Net8_Server OPTIONS (SET DB2_TWO_PHASE_COMMIT 'N');
```

Este ejemplo muestra cómo definir un archivo de rastreo de XA como `D:\Temp\sybase_xa.log` para el derivador Sybase utilizando la sentencia `ALTER SERVER` y la opción de servidor `XA_OPEN_STRING_OPTIONS`:

```
ALTER SERVER Ctlib_Server OPTIONS (ADD XA_OPEN_STRING_OPTIONS
'-LD:\Temp\sybase_xa.log');
```

Este ejemplo muestra cómo inhabilitar temporalmente la confirmación en dos fases mediante la sentencia `SET SERVER OPTION`:

```
SET SERVER OPTION DB2_TWO_PHASE_COMMIT TO 'N' FOR SERVER Net8_Server;
```

Requisitos y configuración del origen de datos para las transacciones de confirmación federada en dos fases

Antes de habilitar la confirmación federada en dos fases para un origen de datos, debe comprobar que sea un origen de datos válido.

Los sistemas federados pueden ejecutar operaciones de confirmación en dos fases con los orígenes de datos siguientes:

- Orígenes de datos DB2 mediante el protocolo Distributed Relational Database Architecture (DRDA):
 - DB2 Universal Database para Linux, UNIX y Windows, versión 8.1 o posterior
 - DB2 Universal Database para z/OS, versión 7.1 o posterior
 - DB2 Universal Database para System i, versión 5.3 o posterior
- Informix IDS, versión 7.31 o posterior, versión 9.40 o posterior, versión 10.0 o posterior
- Informix XPS, versión 8.40 o posterior
- Microsoft SQL Server 2000 y Microsoft SQL Server 2005 para un servidor federado solamente en Windows
- Oracle, versión 8.1.7 o posterior, con la biblioteca XA
- Sybase Adaptive Server Enterprise, versión 12 o posterior, con la biblioteca XA para un servidor federado solamente en Windows

Si intenta habilitar la confirmación federada en dos fases para un origen de datos no compatible, obtendrá un error SQL1881N.

Configuración de orígenes de datos DRDA

El servidor federado proporciona conectividad con orígenes de datos DB2 mediante la utilización del protocolo abierto DRDA. Esta funcionalidad es equivalente a la proporcionada por el servidor DB2 Connect.

Además, para la confirmación en dos fases, el servidor federado interactúa con cada origen de datos utilizando el modelo estándar XA.

Antes de empezar

Restricciones:

- No todos los orígenes de datos DB2 son compatibles de forma nativa con XA sobre DRDA. En los casos en que no son compatibles, como ocurre con DB2 para z/OS y DB2 para System i, el servidor federado utiliza el gestor de puntos de sincronismo (SPM). El gestor de puntos de sincronismo correlaciona los flujos de confirmación en dos fases de XA con flujos que no son de XA y que son compatibles con todos los servidores DB2. Cuando se utiliza el gestor de puntos de sincronismo para la confirmación en dos fases, no se puede utilizar la semántica completa de XA, pues existen incompatibilidades entre el soporte federado y el gestor de puntos de sincronismo. Por ejemplo, las transacciones no se pueden anidar. Todas las transacciones se deben confirmar o retrotraer antes de iniciar una nueva transacción.
- La confirmación federada en dos fases es compatible con DB2 para z/OS, pero DB2 para z/OS no permite la emisión de una sentencia SAVEPOINT en una transacción de confirmación federada en dos fases.
- En una transacción coordinada por DB2, un cliente DB2 para z/OS puede realizar una actualización federada de una fase sobre una conexión entrante de dos fases de DRDA. Sin embargo, esa actualización no se puede efectuar sobre una conexión entrante de dos fases de DRDA para XA. Ni tampoco puede ejecutarse una combinación de actualizaciones de una fase y dos fases sobre una conexión entrante de dos fases de DRDA.

- La opción de servidor XA_OPEN_STRING_OPTIONS no se puede utilizar para orígenes de datos DRDA. Si utiliza esa opción, se devuelve un error SQL1881.

Requisitos:

- Para los orígenes de datos DB2 que son compatibles con XA mediante el uso de SPM en lugar de serlo de forma nativa, compruebe que los parámetros SPM_NAME y SVCENAME estén establecidos correctamente en sus valores por omisión en la configuración del gestor de bases de datos.

Procedimiento

Para configurar un origen de datos DRDA:

Ejecute la sentencia CREATE SERVER, ALTER SERVER o SET SERVER con la opción DB2_TWO_PHASE_COMMIT establecida en S. El derivador DRDA crea automáticamente la siguiente cadena de caracteres de XA OPEN para orígenes de datos DRDA:

```
DB=nombre_base_datos,UID=id_usuario,PWD=contraseña,TPM=FDB2,HOLD_CURSOR=T
```

Configuración de orígenes de datos Oracle

Existen varios requisitos y restricciones respecto a la utilización de orígenes de datos Oracle para la confirmación federada en dos fases.

Antes de empezar

Restricciones:

- El DDL de paso a través y el DDL transparente dirigidos a Oracle producen el error SQL30090 junto con el código de razón 21 (ORA-2089). El SQL normal emitido en sesiones de paso a través funciona de forma correctamente.

Requisitos:

- El cliente Oracle utilizado en el servidor federado debe ser una instalación completa para asegurar que estén presentes todas las bibliotecas referentes a XA. Compruebe que djsxlinkOracle se ha ejecutado satisfactoriamente para que todas las bibliotecas de DB2 y Oracle estén disponibles y enlazadas debidamente. Observe que los scripts djsxlink si ejecutan automáticamente si instala el cliente Oracle antes de instalar el servidor federado.
- Debe otorgar los privilegios siguientes a todos los usuarios que ejecuten transacciones con confirmación en dos fases desde el servidor federado:
 - grant select on dba_pending_transactions to USERID;
 - grant select on dba_2pc_pending to USERID;
 - grant force transaction to USERID;
- Opcionalmente, puede también otorgar el privilegio siguiente a los usuarios que ejecuten transacciones con confirmación en dos fases desde el servidor federado:
 - grant force any transaction to USERID;
- Si piensa ejecutar simultáneamente más de 10 transacciones con confirmación en dos fases, puede ser conveniente aumentar el valor del parámetro distributed_transactions para el servidor Oracle en el archivo init.ora.

Procedimiento

Para configurar un origen de datos Oracle:

1. Ejecute la sentencia CREATE SERVER, ALTER SERVER o SET SERVER con la opción DB2_TWO_PHASE_COMMIT establecida en S.

El derivador Oracle crea automáticamente la siguiente cadena de caracteres XA OPEN para orígenes de datos Oracle:

```
Oracle_XA=Acc=Pid_usuario/contraseña+SesTm=0+DB=nombre_base_datos+SqlNet=enlace_base_datos+Thread
```

Por ejemplo:

```
XA_OPEN_STRING_OPTIONS '+LogDir=/home/user/directory+DbgFl=0x7'
```

2. Puede especificar opciones de XA adicionales mediante la opción de servidor XA_OPEN_STRING_OPTIONS.

Configuración de orígenes de datos Informix

Existen varios requisitos y restricciones respecto a la utilización de orígenes de datos Informix para la confirmación federada en dos fases.

Antes de empezar

Restricciones:

- No puede acceder a apodos Informix utilizando una combinación de un servidor de confirmación en dos fases y un servidor de confirmación en una fase en una misma conexión con un servidor federado.
- La opción de cursor WITH HOLD no está permitida.
- La opción de servidor XA_OPEN_STRING_OPTIONS no se puede utilizar para orígenes de datos Informix.

Requisitos:

- El registro de anotaciones debe estar habilitado para la base de datos Informix.
- La biblioteca XA de Informix solamente permite una sola conexión por hebra. Como resultado, el servidor federado no puede acceder a orígenes de datos Informix utilizando varios servidores que están habilitados para la confirmación federada en dos fases en una misma conexión. Si una aplicación necesita utilizar varios servidores que están habilitados para la confirmación federada en dos fases, siga los pasos opcionales del procedimiento siguiente.

Procedimiento

Para configurar un origen de datos Informix:

1. Ejecute la sentencia CREATE SERVER, ALTER SERVER o SET SERVER con la opción DB2_TWO_PHASE_COMMIT establecida en S.

El derivador Informix crea automáticamente la siguiente cadena de caracteres de XA OPEN para orígenes de datos Informix:

```
DB=nombre_base_datos;RM=nombre_gestor_recursos;CON=con;USER=usuario;PASSWD=contraseña
```

2. Si una aplicación necesita utilizar varios servidores que están habilitados para la confirmación federada en dos fases, siga los pasos siguientes:
 - a. Copie las bibliotecas del derivador Informix: libdb2informix.a, libdb2informixF.a y libdb2informixU.a.
 - b. Defina varias instancias del derivador Informix especificando una copia diferente de las bibliotecas del derivador Informix en la cláusula LIBRARY de la sentencia CREATE SERVER.
 - c. Defina cada servidor con confirmación federada en dos fases para las diferentes instancias de derivador.

Por ejemplo:

```

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 'S');
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 'S');

```

Configuración de orígenes de datos de Microsoft SQL Server

Existen varios requisitos y restricciones respecto a la utilización de orígenes de datos de Microsoft SQL Server para la confirmación federada en dos fases.

Antes de empezar

Restricciones:

- Para los orígenes de datos de Microsoft SQL Server, la confirmación federada en dos fases solamente se puede utilizar con IBM InfoSphere Federation Server instalado en Windows.
- El nivel de aislamiento de DB2 no se propaga a Microsoft SQL Server.

Requisitos:

- Para que la confirmación federada en dos fases sea efectiva con Microsoft SQL, se debe añadir la opción de servidor XA_OPEN_STRING_OPTIONS al servidor:

```
alter server S1 options(add xa_open_string_options
  'RMRecoveryGuid=c200e360-38c5-11ce-ae62-08002b2b79ef');
```

donde RMRecoveryGuid = ID del gestor de recursos.

El ID del gestor de recursos se encuentra en la siguiente ubicación del Registro de Microsoft SQL Server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer]
  "ResourceMgrID" = "{ID de gestor de recursos}"
```

Procedimiento

Para configurar un origen de datos de Microsoft SQL Server:

1. Ejecute la sentencia CREATE SERVER, ALTER SERVER o SET SERVER con la opción DB2_TWO_PHASE_COMMIT establecida en S.
El derivador de Microsoft SQL Server crea automáticamente la siguiente cadena de caracteres de XA OPEN para orígenes de datos de Microsoft SQL Server:
TM=nombre_gestor_transacciones
2. Utilice la opción de servidor XA_OPEN_STRING_OPTIONS para especificar otras opciones de XA además del valor necesario de RMRecovery Guid.

Configuración de orígenes de datos Sybase

Existen varios requisitos y restricciones respecto a la utilización de orígenes de datos Sybase para la confirmación federada en dos fases.

Antes de empezar

Restricciones:

- Para los orígenes de datos de Sybase, la confirmación federada en dos fases solamente se puede utilizar con IBM InfoSphere Federation Server instalado en Windows.

- El DDL de paso a través y el DDL transparente dirigidos a Sybase producen ambos el error SQL910N. Es efectiva la utilización de SQL normal en sesiones de paso a través.

Requisitos:

- El administrador de bases de datos Sybase debe tener una licencia para la gestión de transacciones distribuidas de Adaptive Server Enterprise (ASE) para Sybase y debe habilitar la función con el mandato siguiente en la herramienta isql:

```
sp_configure 'enable dtm', 1
```

 Se debe reiniciar ASE de Sybase para que este parámetro sea efectivo.
- El nombre de usuario especificado en una cadena de caracteres abierta debe tener el rol dtm_tm_role en el correspondiente ASE de Sybase. El usuario administrativo puede asignar este rol con el mandato siguiente en la herramienta isql:

```
sp_role "grant", dtm_tm_role, user_name
```
- Para que un origen de datos de ASE (Adaptive Server Enterprise) de Sybase actúe como gestor de recursos para una base de datos federada, debe existir una entrada para el gestor de recursos lógicos (LRM) en el archivo xa_config del directorio \$SYBASE/\$SYBASE_OCS/config (versión 12 o posterior) que correlacione el nombre del gestor de recursos con el nombre de ASE de Sybase. Consulte la documentación de Sybase ASE XA para obtener más información.
 El nombre del gestor de recursos lógicos es utilizado por el servidor federado en la cadena de caracteres de XA OPEN. El servidor federado utiliza el nombre de nodo de ASE de Sybase para el nombre del gestor de recursos lógicos.
- Compruebe que el nombre de servidor especificado en el archivo de configuración de XA xa_config existe en el archivo de inicialización sql.ini del directorio \$SYBASE/ini.

Procedimiento

Para configurar un origen de datos Sybase:

1. Instale el archivo de biblioteca libxadtm.dll de Sybase XA en el servidor federado.
2. Antes de utilizar las funciones de confirmación federada en dos fases, cree las entradas siguientes para el gestor de recursos lógicos en el archivo \$SYBASE/\$SYBASE_OCS/config/xa_config. Si no tiene permisos de escritura para el archivo xa_config, cree un archivo xa_config en otro directorio y especifique la vía de acceso absoluta del archivo en la variable de entorno XACONFIGFILE en el archivo db2dj.ini:

```
;es necesaria una línea de comentarios
lrm=nombre_gestor_recursos_lógicos
server=nombre_servidor
```

 donde *nombre_servidor* es el nombre de una entrada contenida en el archivo \$SYBASE/ini/sql.ini.
3. El derivador Sybase crea automáticamente por omisión la siguiente cadena de caracteres XA_OPEN para orígenes de datos Sybase:

```
-Nombre_gestor_recursos -Uid_usuario -Pcontraseña
```

Si necesita especificar otras opciones para la cadena de caracteres XA_OPEN, utilice la opción de servidor XA_OPEN_STRING_OPTIONS.

Recuperación de problemas de confirmación federada en dos fases

Si se produce un problema durante una confirmación en dos fases, un sistema federado puede realizar la recuperación mediante la resincronización automática o recuperación manual de las transacciones dudosas.

Resincronización para sistemas federados

La confirmación federada en dos fases incluye un proceso automático de manejo de errores para las transacciones de confirmación.

Además de los errores producidos en el servidor federado, un entorno federado aumenta el riesgo de que se produzcan errores como resultado de anomalías de la red, las comunicaciones o los orígenes de datos.

Para asegurar la integridad de los datos, el servidor federado maneja estos errores durante el proceso de confirmación federada en dos fases:

Error en la primera fase

Si una base de datos comunica que no ha podido prepararse para confirmar una unidad de trabajo, el servidor federado retrotrae la unidad de trabajo durante la segunda fase del proceso de confirmación. Durante la segunda fase, el servidor federado envía un mensaje de retrotracción a todos los orígenes de datos participantes que están a la espera del resultado de la transacción.

Error en la segunda fase

El manejo de errores en esta fase depende de si la segunda fase confirma o retrotrae la transacción. La segunda fase solamente retrotrae la transacción si la primera fase encontró un error.

Si uno de los orígenes de datos participantes no puede confirmar ni retrotraer la unidad de trabajo, posiblemente debido a un error de comunicaciones, el servidor federado reintenta la confirmación o retrotracción mediante un proceso llamado resincronización. La resincronización es iniciada y gestionada por el servidor federado de forma automática. Se informa a la aplicación solicitante de que la confirmación fue satisfactoria mediante la SQLCA (área de comunicaciones de SQL) si la aplicación se conecta al servidor federado mediante una conexión de confirmación en dos fases. La aplicación solicitante se desconecta del servidor federado si la aplicación se conecta al servidor federado mediante una conexión de confirmación en una fase.

La mayoría de los orígenes de datos no son capaces de iniciar una resincronización si se produce un error en un sistema federado. El servidor federado inicia el proceso de resincronización.

En determinados casos, el servidor federado puede fallar durante el proceso de transacciones, por ejemplo, debido a una interrupción del suministro eléctrico. Normalmente la resincronización resuelve cualquier transacción de unidad de trabajo distribuida sin la intervención del usuario.

La resincronización intenta completar todas las transacciones dudosas. Como parte de la resincronización normal, el agente de resincronización se conecta con la base de datos del gestor de recursos para una transacción y emite una decisión de confirmación o retrotracción. A continuación, el gestor de transacciones federadas propaga esa decisión a los orígenes de datos que intervenían en la transacción de unidad de trabajo distribuida.

Recuperación manual de transacciones dudosas

Si no puede esperar a que la resincronización resuelva automáticamente las transacciones dudosas, puede resolverlas manualmente. Este proceso se denomina a veces proceso heurístico.

Por ejemplo, el enlace de comunicaciones entre el gestor de transacciones externo y un gestor de transacciones federado falla durante una transacción. Si tiene información suficiente sobre la transacción, puede liberar recursos en el servidor federado y en orígenes de datos remotos retrotrayendo la transacción desde el servidor federado.

Utilice el proceso heurístico solamente cuando conozca la razón del error de transacción y deba liberar inmediatamente recursos bloqueados. En la mayoría de las situaciones, deje que la resincronización automática recupere las transacciones. Existen varios niveles de gestión de transacciones en un sistema federado. La recuperación heurística de transacciones es un proceso complejo y potencialmente peligroso.

Existen tres formas básicas de realizar un proceso heurístico:

- El mandato LIST INDOUBT TRANSACTIONS
Puede utilizar este mandato desde la línea de mandatos para realizar el proceso heurístico.
- La ventana del Gestor de transacciones dudosas.
Puede utilizar esta herramienta de la interfaz gráfica de usuario para realizar el proceso heurístico.
- Las API heurísticas
Puede utilizar estas API dentro de las aplicaciones para realizar el proceso heurístico.

Las operaciones y tareas específicas que el usuario utiliza para realizar el proceso heurístico varían dependiendo de las circunstancias del error.

En los sistemas federados, cuando se envía una petición de proceso heurístico a un gestor de transacciones federado, la decisión resultante de confirmar o retrotraer debe ser compatible con el estado actual de la transacción dudosa en el servidor federado. De lo contrario, se devuelve un mensaje de error.

El estado de la confirmación federada en dos fases para las transacciones dudosas es ligeramente diferente que la confirmación básica en dos fases de DB2 para las transacciones dudosas:

- El estado (d) significa que la transacción no ha recibido el acuse de recibo de confirmación de uno o más orígenes de datos federados.
- El estado (b) significa que la transacción no ha recibido el acuse de recibo de retrotracción de uno o más orígenes de datos federados.

Si no puede confirmar ni retrotraer satisfactoriamente una transacción cuyo estado sea (d) o (b), puede especificar que no se tengan en cuenta las transacciones utilizando la opción (f). Pero si utiliza la opción (f), todos los registros de la transacción se borran del servidor federado y el usuario debe resolver manualmente los problemas de sincronización que haya en los orígenes de datos participantes. Utilice la opción (f) con precaución y solamente cuando sea

absolutamente necesario, tal como en el caso en que un servidor remoto se detenga de forma anómala o se pierdan las conexiones con servidores remotos y exista una necesidad urgente de liberar recursos.

Nota: Debido a que los estados (d) y (b) son nuevos para WebSphere Federation Server v9.1, no se pueden utilizar con clientes federados pertenecientes a versiones anteriores. Si utiliza un cliente de una versión anterior para recuperar manualmente transacciones dudosas, los estados (d) y (b) se correlacionan con el estado (m), el cual no es un valor exacto. Para evitar el uso de información inexacta al recuperar manualmente transacciones dudosas, utilice un cliente federado de la versión 9.1. Por omisión, el sistema donde se ejecuta WebSphere Federated Server v9.1 incluye el cliente federado v9.1.

Rastreo de los estados de una transacción de unidad de trabajo distribuida entre los orígenes de datos

Si decide resolver manualmente las transacciones dudosas en lugar de permitir que la sincronización las resuelva automáticamente, es esencial un rastreo de las transacciones en el sistema federado. Cuando rastrea una transacción dudosa de unidad de trabajo distribuida, la única forma de determinar el origen u orígenes de datos que fallaron es capturar el XID de la transacción anómala.

Debe buscar ese XID en los gestores de bases de datos correspondientes a todos los orígenes de datos a las que pueda haber accedido un servidor federado como parte de la transacción de unidad de trabajo distribuida.

Para determinar el identificador y el estado de cada transacción que intervenía en la unidad de trabajo distribuida que desea rastrear, emita el mandato LIST INDOUBT TRANSACTIONS en la base de datos de la aplicación, la base de datos federada y en todos los orígenes de datos de la transacción de unidad de trabajo distribuida.

Nota: Puede ser necesario un mandato diferente para cada origen de datos que haga uso del protocolo de confirmación en dos fases. Busque el identificador XID en la documentación sobre mandatos correspondiente al origen de datos específico que utilice.

El gestor de transacciones federadas crea un XID en formato hexadecimal durante el proceso de transacciones. Este XID comienza con el identificador de formato F2PC, que es el número 46325243 en formato hexadecimal. El gestor de transacciones federadas envía el XID a los orígenes de datos. Pero antes de que un servidor federado envíe un XID a un origen de datos, el servidor federado cambia el XID para que se ajuste al formato de XID utilizado por ese origen de datos. Estas modificaciones incluyen actualizar la sección del XID referente a la longitud del calificador de rama y añadir la sección del calificador de rama del XID.

Puede ser necesario comparar los XID entre varios orígenes de datos, por lo que debe conocer qué parte del XID puede compararse con exactitud en el entorno de trabajo al rastrear un XID.

Por ejemplo, el gestor de transacciones es una base de datos DB2. Cuando emite el mandato LIST INDOUBT TRANSACTIONS en la base de datos, se devuelve una representación hexadecimal del XID de la transacción, similar a lo siguiente:

```
463250430000019 000000004739314533463135 2E47453934000000 000000000000E80000
```

Esta cadena consta en realidad de varias partes diferenciadas:

ID de formato	Longitud del identificador de transacción	Longitud del identificador de rama	Identificador de transacción
46325043	00000019	00000000	4739314533463135 2E47453934000000 000000000000E800 00

Los valores listados en la cadena son hexadecimales. Por ejemplo, la longitud del identificador de transacción (hexadecimal 19) representa el valor decimal 25.

Si ese mismo XID se pasa desde un servidor federado que actúa como gestor de transacciones federadas a un origen de datos que actúa como gestor de recursos, se añaden datos a la cadena del XID. Por ejemplo, el XID que se pasa a un gestor de recursos para un origen de datos de un sistema de base de datos DB2 para Windows cambia al formato siguiente:

```
463252430000019 000000014739314533463135 2E47453934000000 000000000000E8000001
```

He aquí esa misma cadena cambiada del XID descompuesta en sus elementos estándar:

ID de formato	Longitud de TID	Longitud del identificador de rama	Identificador de transacción	Calificador de rama
46325043	00000019	00000001	4739314533463135 2E47453934000000 000000000000E800 00	01

Ahora la longitud del calificador de rama tiene un 1, y se ha añadido la sección del calificador de rama. En cambio, el campo del identificador de transacción no ha cambiado. Puede todavía rastrear el XID entre los diferentes orígenes de datos limitando la cadena de búsqueda a la sección del identificador de transacción.

Resolución de problemas de la confirmación federada en dos fases

La resolución de problemas de la confirmación en dos fases es a menudo específica del origen de datos que está causando el problema.

Las aplicaciones deben tratar los códigos de error que indican que una transacción ha excedido el tiempo de espera, específicamente los códigos de error -913 y -918, además de comprobar la existencia de códigos de error -911.

Resolución de problemas de orígenes de datos Oracle

Pruebe los métodos siguientes para resolver problemas de orígenes de datos Oracle.

- Para escribir información en un archivo de registro:

```
db2 "alter server ora1 options (add XA_OPEN_STRING_OPTIONS  
'+LogDir=C:\temp+DbgFl=0x7')
```

Donde `C:\temp` es la vía de acceso completa del lugar donde desea crear el archivo de registro. La escritura de información en un archivo de registro puede reducir significativamente la velocidad de proceso, por lo que solamente debe utilizar esa opción para la resolución de problemas.

- Para visualizar información de rastreo, añada las líneas siguientes al archivo `sqlnet.ora` del cliente Oracle.

```
TRACE_LEVEL_CLIENT=16
TRACE_DIRECTORY_CLIENT=C:\temp
```

Puede también asignar un valor más bajo a `TRACE_LEVEL_CLIENT`, tal como 4 u 8. La visualización de información de rastreo puede disminuir significativamente la velocidad de proceso, por lo que solamente debe habilitar la información de nivel de rastreo para la resolución de problemas.

- Para listar las transacciones pendientes existentes en el origen de datos Oracle:

```
select * from dba_pending_transactions where formatid=1177702467;
select * from dba_2pc_pending;
```

Para listar el estado y el ID de una transacción:

```
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;
```

- Para resolver transacciones manualmente:

```
rollback force '4.31.157818';
commit force '10.24.154537'
```

donde `'4.31.157818'` es el campo `A.LOCAL_TRAN_ID` de Oracle que corresponde al `XID` en `GLOBAL_TRAN_ID`.

Resolución de problemas de orígenes de datos Sybase

Pruebe los métodos siguientes para resolver problemas de orígenes de datos Sybase.

- Examine el archivo de registro `syb_xa_log` de Sybase XA, situado en el directorio `$SYBASE`.
- Utilice la herramienta `db2diag` para examinar el archivo `db2diag.log`.
- Para examinar una transacción en el servidor Sybase:

```
$ isql -Unombre_usuario -Pcontraseña -Snombre_servidor
1> sp_transactions
2> go
```

Si encuentra una transacción no válida o innecesaria, solicite al administrador de Sybase que suprima la transacción.

Rendimiento de la confirmación federada en dos fases

Los orígenes de datos que están configurados para transacciones con confirmación en dos fases tienen un rendimiento menor en comparación con los orígenes de datos que están configurados para transacciones con confirmación en una fase.

Cuando un origen de datos utiliza la confirmación en dos fases, el servidor federado actúa como coordinador para asegurar que todos los participantes estén sincronizados correctamente. Esta coordinación se consigue mediante una actividad adicional de registro de anotaciones en el servidor federado y una mayor

comunicación entre orígenes de datos. Como tal, una transacción federada que acceda a un origen de datos para una confirmación en dos fases requiere más proceso que una transacción que acceda a un origen de datos para una confirmación en una fase. En consecuencia, un origen de datos se debe habilitar para la confirmación en dos fases solamente cuando las transacciones federadas requieran una confirmación en dos fases.

Una transacción federada que solamente necesite una confirmación en una fase, tal como una actualización simple, pero que se ejecute en un origen de datos para la confirmación en dos fases probablemente tendrá un rendimiento menor respecto a la misma transacción ejecutada en un origen de datos sin confirmación en dos fases.

Una transacción sometida al control de la confirmación en dos fases produce el proceso adicional siguiente con independencia de si la transacción requiere o no confirmación en dos fases:

1. Todas las transacciones requieren operaciones adicionales de escritura en el archivo de registro del servidor federado.
Las operaciones adicionales de escritura en el archivo de registro permiten al servidor federado supervisar los orígenes de datos de la transacción para coordinar las operaciones subsiguientes de confirmación y retroacción.
2. Todas las transacciones requieren una mayor comunicación entre el servidor federado y el origen de datos.
3. Las operaciones insert, update y delete requieren una o más operaciones adicionales de escritura en el archivo de registro de la base de datos para el origen de datos remoto.

La mayor parte del proceso adicional se realiza a nivel de transacción y no está influido por el contenido de la transacción. Como resultado, el incremento porcentual en el tiempo transcurrido para una transacción corta es mayor que para una transacción larga.

Las transacciones simultáneas hacen que el servidor federado escriba varias entradas en el archivo de registro en una sola operación de escritura. Como consecuencia, las aplicaciones que ejecutan transacciones simultáneas con confirmación en dos fases producen menos actividad adicional que las que ejecutan esas mismas transacciones de forma secuencial.

Mejora del rendimiento de la confirmación federada en dos fases

Puede mejorar el rendimiento de las transacciones en un entorno donde se utilice la confirmación federada en dos fases.

Considere las configuraciones siguientes posibles:

- Para disminuir el tiempo empleado en escribir registros de anotaciones adicionales en el servidor federado, coloque los archivos de registro de la base de datos federada en un dispositivo capaz de realizar transacciones de escritura rápidas, preferiblemente un dispositivo que tenga una antememoria de escritura. Generalmente, las transacciones de escritura en el archivo de registro del servidor federado son las responsables de la mayor parte del tiempo de proceso adicional. El disponer los archivos de registro en una ubicación apropiada probablemente mejorará el rendimiento de la confirmación en dos fases. Esta mejora es especialmente cierta para las transacciones de sólo lectura. La

ubicación de los archivos de registro del servidor federado está determinada por el parámetro de configuración NEWLOGPATH de la base de datos.

- Para las transacciones insert, update y delete, coloque los archivos de registro del origen de datos remoto en un dispositivo capaz de realizar operaciones de escritura rápidas.
- Para disminuir la actividad general producida por los mensajes adicionales de XA que se envían entre el servidor federado y los orígenes de datos:
 - Coloque el servidor federado en la misma máquina que uno de los orígenes de datos de confirmación en dos fases.
 - Si no puede colocar el servidor federado en la misma máquina que un origen de datos, aumente la velocidad de la red y reduzca el tiempo de latencia entre el servidor federado y los orígenes de datos para ayudar a mejorar el rendimiento.

Para las aplicaciones, solamente habilite la confirmación en dos fases para un origen de datos cuando al menos una transacción de la aplicación requiera confirmación en dos fases. Para cada servidor, establezca el valor por omisión de DB2_TWO_PHASE_COMMIT en N y utilice la sentencia SET SERVER OPTION para habilitar la confirmación en dos fases dentro de las aplicaciones que requieran específicamente confirmación en dos fases.

Capítulo 10. Insertar, actualizar y suprimir datos en un sistema federado

Durante el desarrollo del entorno federado, tanto si modifica la información remota o mueve datos entre orígenes de datos, debe insertar, actualizar y suprimir datos en los orígenes de datos.

Antes de modificar los datos en un origen de datos remoto, asegúrese de que dispone de los privilegios de autorización apropiados para emitir las sentencias INSERT, UPDATE y DELETE en el apodo. También debería entender cómo funciona la integridad referencial, preservando la atomicidad de sentencias, y la asignación de la semántica.

Privilegios de autorización para sentencias INSERT, UPDATE y DELETE

Los privilegios necesarios para emitir sentencias INSERT, UPDATE y DELETE sobre apodos son similares a los privilegios necesarios para emitir estas mismas sentencias sobre tablas. Además, el usuario debe tener los privilegios necesarios sobre el origen de datos para realizar operaciones de selección, inserción, actualización y supresión de objetos.

Puede otorgar o revocar privilegios SELECT, INSERT, UPDATE y DELETE para un apodo.

Pero el otorgar o revocar privilegios para un apodo no otorga ni revoca privilegios en el origen de datos. En el origen de datos, los privilegios se deben otorgar o revocar para el ID de autorización remoto (REMOTE_AUTHID) especificado en la correlación de usuarios en el servidor federado.

Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir los privilegios necesarios sobre el apodo (para que la base de datos acepte la petición). El ID de usuario del origen de datos que está correlacionado con el ID de autorización (mediante una correlación de usuarios) debe tener los privilegios necesarios sobre el objeto de tabla (para que el origen de datos acepte la petición).

Cuando se envía una consulta a la base de datos federada, se comprueban los privilegios de autorización sobre el apodo contenidos en la consulta. Los requisitos de autorización del objeto del origen de datos especificado por el apodo solamente se aplican cuando la consulta se procesa realmente. Es necesario tener el privilegio SELECT sobre el apodo para seleccionar el objeto del origen de datos especificado por el apodo.

De la misma manera, no es suficiente tener el privilegio UPDATE sobre el apodo para poder actualizar el objeto del origen de datos representado por el apodo. Una comprobación satisfactoria de los privilegios en el servidor federado no implica que esa comprobación será también satisfactoria en el origen de datos remoto. Mediante correlaciones de usuarios, un ID de autorización del servidor federado se correlaciona con el ID de usuario del origen de datos. La comprobación de privilegios se aplica en el origen de datos.

Restricciones sobre INSERT, UPDATE y DELETE en un sistema federado

Al utilizar sentencias INSERT, UPDATE o DELETE en un sistema federado, se aplican algunas restricciones.

Son aplicables las restricciones siguientes a las actualizaciones realizadas sobre apodos:

- Un objeto de origen de datos que sea de sólo lectura, tal como una vista JOIN, no se puede actualizar
- No puede realizar operaciones de inserción, actualización ni supresión en vistas federadas que se han creado con sentencias UNION ALL. Las vistas federadas que se han creado con sentencias UNION ALL son vistas de sólo lectura.
- Una operación de inserción, actualización o supresión federada o una llamada a un procedimiento federado con una indicación de acceso a datos SQL MODIFIES SQL DATA no es válida en una función, una referencia de tabla de cambio de datos, una sentencia compuesta que especifique ATOMIC (con la excepción de los orígenes de datos DB2 para Linux, UNIX y Windows), un desencadenante y un entorno de ejecución de aplicaciones en los que se cumpla una de las siguientes condiciones:
 - Esté en vigor SAVEPOINT (con la excepción de orígenes de datos DB2 para Linux, UNIX y Windows)
 - Se utilice un cursor desplazable
 - La vista de destino contenga varias tablas o apodos

Orígenes de datos no soportados

El sistema federado no puede utilizar operaciones de inserción, actualización ni supresión para apodos situados en orígenes de datos no relacionales.

Los orígenes de datos siguientes no se pueden utilizar en un sistema federado:

- BioRS
- Excel
- Archivos con estructura de tabla
- Servicios Web
- XML

Integridad referencial en un sistema federado

En un sistema federado, la base de datos federada no aplica la integridad referencial entre los orígenes de datos.

Sin embargo, las restricciones de integridad referencial existentes en un origen de datos pueden afectar a las actualizaciones de apodos. Suponga que necesita insertar datos en un apodo que residen en el servidor federado. Cuando el servidor federado envía la sentencia de inserción al origen de datos, vulnera una restricción de integridad referencial en ese origen de datos. El servidor federado correlaciona el error resultante con un error federado.

Las aplicaciones son las encargadas de mantener la integridad referencial entre los orígenes de datos.

Sentencias INSERT, UPDATE y DELETE y objetos grandes (LOB)

Puede realizar operaciones de lectura sobre objetos LOB remotos en sistemas federados. Las operaciones de escritura sobre objetos LOB están permitidas para algunos orígenes de datos.

En los sistemas federados puede realizar operaciones de lectura sobre objetos LOB que están situados en un origen de datos relacional cualquiera. Puede realizar operaciones de escritura en objetos LOB situados en los orígenes de datos siguientes:

- Oracle, mediante el derivador NET8
- DB2 para z/OS, DB2 para System i y DB2 Database para Linux, UNIX y Windows, mediante el derivador DRDA

En determinadas condiciones, puede realizar operaciones de escritura en objetos LOB situados en otros orígenes de datos cambiando el tipo de columna del apodo a VARCHAR.

Preservación de la atomicidad de las sentencias en un sistema federado

Durante las operaciones de actualización, los sistemas federados intentan siempre mantener los datos en un estado atómico al concluir una sentencia de DML. Cuando los datos están en un estado atómico, se asegura que el proceso de los datos concluya satisfactoriamente o que permanezcan inalterados.

Cuando un cliente o aplicación emite una sentencia INSERT, UPDATE o DELETE sobre un apodo, un servidor federado procesa internamente esa sentencia como sentencia de DML individual o como serie de sentencias de DML. Si un servidor federado debe enviar varias sentencias de DML a un origen de datos para su proceso, la atomicidad de los datos puede estar en peligro. Para evitar poner en peligro la atomicidad de los datos, los sistemas federados utilizan interfaces API de punto de rescate de datos para supervisar una serie de sentencias de DML.

Algunos orígenes de datos no externalizan las API de punto de rescate que puedan ser utilizadas por el sistema federado. En estas condiciones, las sentencias federadas INSERT, UPDATE o DELETE se ejecutan sin la protección proporcionada por las API de punto de rescate.

Cuando se produce un error durante una transacción federada de inserción, actualización o supresión, pueden obtenerse resultados de actualización parciales en los orígenes de datos. Para corregir los problemas de incoherencia, un sistema federado realiza automáticamente una retrotracción de transacciones antes de devolver un error SQLCODE a las aplicaciones.

Los orígenes de datos siguientes no externalizan las API de punto de rescate que puedan ser utilizadas por el servidor federado:

- DB2 for System i
- DB2 for VM and VSE
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Teradata

Cuando se envía una transacción completa de inserción, actualización o supresión al origen de datos para su proceso, el servidor federado supone que el origen de datos conservará la atomicidad de la sentencia si se produce un error. Cuando solamente se envía una parte de la transacción de inserción, actualización o supresión al origen de datos para su proceso, se retrotrae la transacción completa si se produce un error.

Modificación de datos en un sistema federado

Para modificar los datos en un sistema federado, puede insertar, actualizar y suprimir datos de los objetos del origen de datos.

Inserción de datos en objetos de origen de datos

Para insertar datos en orígenes de datos, utilice los apodos de los objetos de origen de datos en la sentencia INSERT.

Para insertar datos utilizando un apodo, se deben cumplir estas condiciones:

- Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir el privilegio INSERT para el apodo (para que la base de datos federada acepte la petición).
- El ID de usuario contenido en el origen de datos debe tener el privilegio INSERT para el objeto de tabla asociado (para que el origen de datos acepte la petición).
- El ID de usuario del origen de datos se debe correlacionar con el ID de autorización del servidor federado mediante una correlación de usuarios.

Restricciones

La federación no es compatible con operaciones INSERT utilizadas con orígenes de datos no relacionales.

Procedimiento

Para insertar datos en objetos de origen de datos, emita la sentencia INSERT.

Ejemplo: una tabla Informix consta de dos columnas. La primera columna contiene datos de tipo INTEGER y la segunda columna contiene datos de tipo VARCHAR (hasta un máximo de 20 caracteres). El apodo *infx_table_nn* está registrado en el servidor federado para la tabla Informix.

Puede emitir sentencias INSERT, UPDATE y DELETE para la tabla Informix utilizando el apodo *infx_table_nn*. La sentencia siguiente inserta una nueva fila de información en la tabla Informix:

```
INSERT INTO db2user1.infx_table_nn VALUES(1,'Walter')
```

Actualización de datos en objetos de origen de datos

Para actualizar datos en orígenes de datos, utilice los apodos de los objetos de origen de datos en la sentencia UPDATE.

Antes de empezar

Para actualizar datos utilizando un apodo, se deben cumplir estas condiciones:

- Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir el privilegio UPDATE para el apodo (para que la base de datos federada acepte la petición).

- El ID de usuario contenido en el origen de datos debe tener el privilegio UPDATE para el objeto de tabla asociado (para que el origen de datos acepte la petición).
- El ID de usuario del origen de datos se debe correlacionar con el ID de autorización del servidor federado mediante una correlación de usuarios.

Restricciones

La federación no es compatible con operaciones UPDATE para algunos orígenes de datos; consulte “Restricciones sobre INSERT, UPDATE y DELETE en un sistema federado” en la página 144.

Para actualizar datos en objetos de origen de datos, emita la sentencia UPDATE.

Ejemplo: una tabla Informix consta de dos columnas. La primera columna contiene datos de tipo INTEGER y la segunda columna contiene datos de tipo VARCHAR (hasta un máximo de 20 caracteres). El apodo *infx_table_nn* está registrado en el servidor federado para la tabla Informix.

Puede emitir sentencias INSERT, UPDATE y DELETE para la tabla Informix utilizando el apodo *infx_table_nn*. La sentencia siguiente actualiza una fila de información en la tabla Informix:

```
UPDATE db2user1.infx_table_nn SET c2='Bill' WHERE c1=2
```

Supresión de datos de objetos de origen de datos

Para suprimir datos de orígenes de datos, utilice los apodos de los objetos de origen de datos en la sentencia DELETE.

Antes de empezar

Para suprimir datos utilizando un apodo, se deben cumplir estas condiciones:

- Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir el privilegio DELETE para el apodo (para que la base de datos federada acepte la petición)
- El ID de usuario contenido en el origen de datos debe tener el privilegio DELETE para el objeto de tabla asociado (para que el origen de datos acepte la petición)
- El ID de usuario del origen de datos se debe correlacionar con el ID de autorización del servidor federado mediante una correlación de usuarios.

Restricciones

La federación no es compatible con operaciones DELETE para algunos orígenes de datos; consulte “Restricciones sobre INSERT, UPDATE y DELETE en un sistema federado” en la página 144.

Procedimiento

Para suprimir datos en objetos de origen de datos, emita la sentencia DELETE.

Ejemplo: una tabla Informix consta de dos columnas. La primera columna contiene datos de tipo INTEGER y la segunda columna contiene datos de tipo VARCHAR (hasta un máximo de 20 caracteres). El apodo *infx_table_nn* está registrado en el servidor federado para la tabla Informix.

Puede emitir sentencias INSERT, UPDATE y DELETE para la tabla Informix utilizando el apodo *infx_table_nn*. La sentencia siguiente suprime una fila de información en la tabla Informix:

```
DELETE FROM infx_table_nn WHERE c1=3
```

Semántica de asignación en un sistema federado

Cuando asigna datos a una columna de apodo, el tipo de datos puede cambiar de acuerdo con las reglas de asignación utilizadas por el sistema federado. Es conveniente que comprenda las reglas de asignación para obtener los resultados previstos.

Las reglas para determinar el tipo de datos que se debe asignar a una columna de apodo son las siguientes:

- Determine el tipo de fuente local: el tipo de fuente local está determinado por el tipo de columna local o el tipo de resultado local de las expresiones. Si el fuente es constante, el tipo de fuente local es constante.
- Determine el tipo de destino
 - Si el origen de asignación carece de tipo, tal como los marcadores de parámetros y los valores NULL, el tipo de destino es MIN(tipo_destino_local, tipo_destino_remoto), donde tipo_destino_local es el tipo de datos local de la columna actualizada y tipo_destino_remoto es el tipo de datos del origen de datos de la columna actualizada. Tipo_destino_remoto representa el tipo de correlación de tipos por omisión del tipo de datos de la columna de destino remota.
 - Si el fuente de asignación no es NULL ni marcadores de parámetros, el tipo de destino es MIN(tipo_destino_local, tipo_destino_remoto, tipo_fuente_local).

La definición de MIN(tipo1, tipo2)

- Tipo1 y tipo2 no son exactamente iguales.
- MIN(tipo1,tipo2) = MIN(tipo2, tipo1)
- MIN(tipo1, tipo2) = tipo_destino_remoto(tipo_destino_local), when MIN(tipo1, tipo2) = DECIMAL(0,0)
- BLOB solamente es compatible con BLOB, por tanto MIN(BLOB(x), BLOB(y))=BLOB(z) donde z=min(x,y)
- Los tipos de datos TIME y DATE no son compatibles.
- Los tipos Datetime y las series de caracteres son compatibles.
- En las bases de datos Unicode, las series de caracteres y las series gráficas son compatibles.

Las tablas siguientes muestran el mínimo de dos tipos de datos para los tipos de datos numérico, serie de caracteres, serie gráfica, y fecha y hora.

Tabla 10. Tipos de datos numéricos

tipo1	tipo2	MIN(tipo1, tipo2)
SMALLINT	SMALLINT o INTEGER o BIGINT o REAL o DOUBLE	SMALLINT
INTEGER	BIGINT o REAL o DOUBLE	INTEGER
BIGINT	REAL o DOUBLE	BIGINT
REAL	DOUBLE	REAL

Tabla 10. Tipos de datos numéricos (continuación)

tipo1	tipo2	MIN(tipo1, tipo2)
DECIMAL(w,x)	SMALLINT	DECIMAL(p,0) donde $p=w-x$, si $p < 5$; SMALLINT, en otro caso
DECIMAL(w,x)	INTEGER	DECIMAL(p,0) donde $p=w-x$, si $p < 11$; INTEGER, en otro caso
DECIMAL(w,x)	BIGINT	DECIMAL(p,0) donde $p=w-x$, si $p < 19$; BIGINT, en otro caso
DECIMAL(w,x)	DECIMAL(y,z)	DECIMAL(p,s) donde $p = \min(w,y) + \min(w-x,y-z)$, $s = \min(x,z)$
DECIMAL(w,x)	DOUBLE o REAL	DECIMAL(w,x)

La tabla siguiente muestra el mínimo de dos tipos de datos para los tipos de datos de serie de caracteres.

Tabla 11. Tipos de datos de serie de caracteres

tipo1	tipo2	MIN(tipo1, tipo2)
CHAR(x)	CHAR(y) o VARCHAR(y) o LONG VARCHAR o CLOB(y)	CHAR(z) donde $z = \min(x,y)$
VARCHAR(x)	VARCHAR(y) o LONG VARCHAR o CLOB(y)	VARCHAR(z) donde $z = \min(x,y)$
LONG VARCHAR	CLOB(y)	LONG VARCHAR donde $x > 32700$, CLOB(x) donde $x \leq 32700$
CLOB(x)	CLOB(y)	CLOB(z) donde $z = \min(x,y)$

La tabla siguiente muestra el mínimo de dos tipos de datos para los tipos de datos de serie gráfica.

Tabla 12. Tipos de datos de serie gráfica

tipo1	tipo2	MIN(tipo1, tipo2)
GRAPHIC(x)	GRAPHIC(y) o VARGRAPHIC(y) o LONG VARGRAPHIC o DBCLOB(y)	GRAPHIC(z) donde $z = \min(x,y)$
VARGRAPHIC(x)	VARGRAPHIC(y) o LONG VARGRAPHIC o DBCLOB(y)	VARGRAPHIC(z) donde $z = \min(x,y)$
LONG VARGRAPHIC	DBCLOB(y)	LONG VARGRAPHIC donde $x > 32700$, DBCLOB(x) donde $x \leq 32700$
DBCLOB(x)	DBCLOB(y)	DBCLOB(z) donde $z = \min(x,y)$

La tabla siguiente muestra el mínimo de dos tipos de datos para los tipos de datos de fecha y hora.

Tabla 13. Tipos de datos de fecha y hora

tipo1	tipo2	MIN(tipo1, tipo2)
DATE	TIMESTAMP	DATE
TIME	TIMESTAMP	TIME

Si los datos de tipo CHAR que está insertando son más cortos que la longitud de destino, el origen de datos rellena el resto de la columna.

Si está insertando datos de tipo DATE o TIME en una columna remota del tipo de datos TIMESTAMP, el origen de datos rellena el resto de la columna.

Semántica de asignación en un sistema federado - ejemplos

La tabla siguiente muestra varios ejemplos de la aplicación de la semántica de asignación federada en consultas a las que se asigna un tipo local y un tipo remoto.

Tabla 14. Ejemplos de semántica de asignación

Tipo local	Tipo remoto	Consulta realizada por usuario	Consulta remota creada
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' es de tipo VARCHAR(3) y es el más corto de todos)
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 -- si expr1 devuelve char(n) n<= 10 • set c1=CHAR(expr1, 10) -- si expr1 devuelve char(n) n>10
TIMESTAMP	DATE	set c1= current timestamp	set c1=DATE(current timestamp)

Restricciones de orígenes de datos para los valores de tipos de datos

Para algunos tipos de datos, el servidor federado da soporte a una gama más amplia de valores que el origen de datos.

Cuando utilice estos tipos de datos, utilice los valores a los que sólo da soporte el origen de datos. Si inserta un valor que se encuentra fuera del rango admitido, el origen de datos convierte el valor o devuelve un error.

Si utiliza un predicado que contiene un valor que se encuentra fuera del rango admitido y el predicado se evalúa en el origen de datos, puede producirse uno de los errores siguientes:

- El predicado puede devolver un resultado distinto que el mismo predicado en el servidor federado.
- El predicado puede dar lugar a un error.

Horas e indicaciones de fecha y hora con 24 en el campo de horas

Si utiliza un predicado con una hora o indicación de fecha y hora que contiene el valor 24 en el campo de horas, es posible que reciba un error. Para evitar problemas, puede convertir estas horas o indicaciones de fecha y hora: Puede utilizar una sentencia parecida a la sentencia UPDATE siguiente:

```
UPDATE my_table SET timestamp_col = timestamp_col + 0 SECONDS;
```

Esta actualización cambia el campo de hora por 00:00:00 e incrementa el día en un día.

El servidor federado permite que las horas e indicaciones de fecha y hora contengan el valor 24 en el campo de horas. La diferencia entre una indicación de fecha y hora con el valor de 24 horas y una indicación de fecha y hora con el valor de 00 horas es que son distintas en las comparaciones, pero iguales en cuanto a aritmética.

Ejemplo

- El predicado siguiente devuelve el valor false:
`2008-05-14-24:00:00.000000 = 2008-05-15-00:00:00.000000`
- El predicado siguiente devuelve el valor true:
`(2008-05-14-24:00:00.000000 + 0 SECONDS) =
(2008-05-15-00:00:00.000000 + 0 SECONDS)`

Algunos orígenes de datos tales como Oracle y Sybase no permiten el uso de horas o indicaciones de la fecha y hora con el valor 24 en el campo de horas.

Series vacías en Oracle

Para evitar los problemas relacionados con series vacías, no utilice series vacías en aplicaciones que utilicen apodos de Oracle. En lugar de ello, puede utilizar el valor NULL o una serie formada por un único espacio en blanco (' ').

En columnas de tipo VARCHAR de un servidor federado que no sea compatible con VARCHAR2, se hace una distinción entre la serie vacía y los valores NULL, lo cual da como resultado comportamientos diferentes entre las tablas locales y los apodos. En cambio, en columnas de tipo VARCHAR de un origen de datos remoto compatible con VARCHAR2, la serie vacía y los valores NULL se consideran equivalentes. Si inserta una serie vacía ('') en una columna VARCHAR, se convierte en un valor NULL.

En los ejemplos siguiente, MY_TABLE es una tabla local de la base de datos federada y MY_NICK es un apodo de una tabla remota de una base de datos Oracle:

Ejemplo 1

En este ejemplo, se inserta una serie vacía en MY_TABLE y MY_NICK, que contienen ambos una columna de tipo NOT NULL denominada NOT_NULL_COL. La sentencia INSERT de MY_TABLE es satisfactoria, pero la sentencia INSERT de MY_NICK falla con un error debido a que la serie vacía se convierte en un valor NULL, que entra en conflicto con la restricción NOT NULL.

```
INSERT INTO my_table(not_null_col) VALUES('');  
INSERT INTO my_nick(not_null_col) VALUES('');
```

Ejemplo 2

En este ejemplo, tanto el servidor federado como las tablas remotas están vacíos inicialmente. La primera sentencia SELECT no devuelve filas porque el servidor federado distingue entre la serie vacía y los valores NULL, pero la segunda sentencia SELECT devuelve una fila porque, durante la operación insert, la base de datos Oracle convierte la serie vacía en un valor 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;

```

Actualizaciones federadas con puntos de rescate de aplicación

Los puntos de rescate de aplicación en un sistema federado ayudan a controlar el trabajo realizado por un subconjunto de sentencias SQL en una transacción.

Puede utilizar múltiples puntos de rescate dentro de una única transacción. Un subconjunto de sentencias SQL agrupadas bajo un punto de rescate pueden tratarse como una unidad. Se puede dividir de forma lógica una transacción en un único nivel o en niveles anidados de unidades de punto de rescate.

Después de haber establecido un punto de rescate dentro de la aplicación, se puede liberar el punto de rescate o retrotraer el trabajo realizado desde que se ha establecido el punto de rescate. Por ejemplo, si una sentencia SQL individual dentro de un punto de rescate falla, la unidad permanecerá intacta. Puede realizar una de las acciones siguientes:

- Deshacer todas las sentencias SQL ejecutadas dentro del punto de rescate retrocediendo hasta el punto de rescate.
- Liberar el punto de rescate cuando ya no haya necesidad de mantenerlo en el sistema.

Puede realizar actualizaciones federadas con operaciones de punto de rescate de aplicación en DB2 Database para Linux, UNIX y Windows.

Actualizaciones federadas con puntos de rescate de aplicación - ejemplos

Este ejemplo de puntos de rescate de aplicación ilustra el efecto de las operaciones de retrotracción dentro de los puntos de rescate anidados.

Ejemplo: cree un apodo llamado NN_DEPT para la tabla DEPARTMENT en el esquema DEPTDATA en el servidor DB2_SERVER. El apodo NN_DEPT contiene las columnas DEPT_NO, DEPT_NAME y MANAGER_NO.

```
CREATE NICKNAME NN_DEPT FOR DB2_SERVER.DEPTDATA.DEPARTMENT;
```

Inserte una fila antes de crear el punto de recuperación SP1. Inserte otra fila antes de crear el punto de recuperación SP2. Inserte dos filas más antes de crear el punto de recuperación SP3. Inserte una quinta fila.

```

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');

```

El apodo NN_DEPT contendrá cinco filas con los departamentos A10, B10, C10, C20 y D10. Desea retrotraer algunas de estas filas. Los ejemplos siguientes describen el resultado de retrotraer los puntos de recuperación:

- ROLLBACK TO SAVEPOINT SP3;

La última fila con el departamento D10 ya no está en el apodo NN_DEPT. Las filas insertadas antes de crear el punto de recuperación SP3 (A10, B10, C10, C20) aún existen en el apodo NN_DEPT.

- ROLLBACK TO SAVEPOINT SP2;

Las filas con los departamentos C10, C20, D10 ya no están en el apodo NN_DEPT. Las filas insertadas antes de crear el punto de recuperación SP2 (A10, B10) aún existen en el apodo NN_DEPT.

- ROLLBACK TO SAVEPOINT SP1;

Las filas con los departamentos B10, C10, C20 y D10 ya no están en el apodo NN_DEPT. La fila insertada antes de crear el punto de recuperación SP1 (A10) aún existe en el apodo NN_DEPT.

Restricciones sobre las actualizaciones federadas con puntos de rescate de aplicación

Los sistemas federados imponen restricciones sobre las operaciones de puntos de rescate.

El soporte de punto de rescate para las operaciones de grabación en aplicaciones federadas se limita a DB2 Database para Linux, UNIX y Windows. Las restricciones siguientes se aplican al objeto de origen de datos para el cual se ha definido el apodo (como destino de la operación de grabación):

- Para la Versión 9.5, el objeto de origen de datos en una operación de punto de rescate puede ser el propio apodo.
- Para versiones anteriores, el objeto de origen de datos en una operación de punto de rescate no puede ser un apodo.

Puede activar puntos de rescate en modalidad en serie, el entorno de proceso paralelo masivo (MPP) y la confirmación en una fase federada o entornos de confirmación en dos fases. Cualquier restricción que exista en estos entornos también se aplicará a las operaciones de punto de rescate.

Capítulo 11. Importación y exportación de datos para apodos

Puede utilizar el mandato `IMPORT` para importar datos a un apodo y el mandato `EXPORT` para exportar datos desde una consulta donde se especifica un apodo.

Puede utilizar el mandato `IMPORT` solamente con los orígenes de datos siguientes:

- Familia de productos DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

Restricciones al importar datos en apodos

Existen restricciones respecto a la utilización del mandato `IMPORT` para importar datos a un apodo.

Son aplicables las restricciones siguientes al utilizar el mandato `IMPORT` para importar datos a un apodo:

- El objeto remoto sobre el que se define el apodo debe ser una tabla. No puede importar datos a un apodo que esté definido sobre una vista o sinónimo.
- Los tipos de archivo válidos son IXF, ASC y DEL.
- Se debe especificar la cláusula `ALLOW WRITE ACCESS`. Esta cláusula hace que se utilice la modalidad de importación en línea. La cláusula `ALLOW WRITE ACCESS` permite el acceso de lectura y escritura a la tabla de destino de la importación para varias aplicaciones.
- No puede utilizar la modalidad `COMMITCOUNT AUTOMATIC` con apodos.
- Se debe especificar `COMMITCOUNT n`, donde *n* es un número válido distinto de cero.
- Solamente se puede utilizar operaciones `INSERT` e `INSERT_UPDATE` con apodos.
- El tipo de columna `LOB` y las columnas generadas no son válidos para apodos. Para importar datos de tipo `LOB` a una tabla remota, el tipo de datos de la columna del apodo correspondiente debe ser `VARCHAR`.
- Los modificadores `filetype` siguientes no se pueden utilizar con apodos:
 - `dldelfiletype`
 - `generatedignore`
 - `generatedmissing`
 - `identityignore`
 - `identitymissing`
 - `indexixf`
 - `indexschema`
 - `lobsinfile`
 - `nodefaults`
 - `no_type_idfiletype`
 - `usedefaults`
- La jerarquía (tabla con tipo) no es compatible con el uso de apodos.

Si emite un mandato `IMPORT` que no se ajusta a estas restricciones, se devuelve el código de error -27999N de SQL. Por ejemplo:

SQL27999N - La operación `IMPORT` solicitada para un destino remoto (apodo) no puede realizarse. Código de razón = "código_razón"

Utilización del mandato `IMPORT` con apodos - ejemplos

Los ejemplos le muestran cómo importar datos hacia apodos a partir de tipos de archivo diferentes.

Tipo de archivo `DEL`

Este ejemplo utiliza la opción `INSERT_UPDATE` para importar datos desde el tipo de archivo `DEL`:

```
IMPORT FROM import_file_1.del OF DEL
  ALLOW WRITE ACCESS
  COMMITCOUNT 50
  INSERT_UPDATE INTO NICKNAME_1;
```

Tipo de archivo `IXF`

Este ejemplo utiliza la opción `INSERT` para importar datos desde el tipo de archivo `IXF`:

```
IMPORT FROM import_file_1.ixf OF IXF
  ALLOW WRITE ACCESS
  COMMITCOUNT 20
  INSERT INTO NICKNAME_1;
```

Tipo de archivo `ASC`

Este ejemplo utiliza la opción `INSERT` para importar datos desde el tipo de archivo `ASC`. El ejemplo incluye el modificador de archivo `STRIPTBLANKS` para truncar los espacios de cola que haya en los datos. El parámetro `METHOD L` especifica el comienzo y final de las columnas de números.

```
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;
```

Restricciones para exportar datos utilizando apodos

Existen restricciones sobre la utilización del mandato `EXPORT` para exportar datos desde una consulta donde se especifica un apodo.

Son aplicables las restricciones siguientes al utilizar el mandato `EXPORT` para exportar datos utilizando un apodo:

- La descripción de la tabla de destino que es necesaria para realizar la operación `CREATE` de importación no se guarda en el formato de archivo `IXF`. Utilice el programa `db2look` para recoger la información que necesite para reconstruir la tabla.
- Puede exportar datos a los tipos de datos `IXF` y `DEL`. El tipo de archivo `ASC` no se puede utilizar para exportar datos de apodos.

Capítulo 12. Trabajar con apodos

Un apodo es un identificador que una aplicación utiliza para hacer referencia a un objeto de origen de datos, como por ejemplo una tabla o vista. En un sistema federado, puede utilizar los apodos para acceder a objetos de origen de datos y mejorar el rendimiento de las consultas en orígenes de datos remotos.

Apodos en un sistema federado

Cuando desee seleccionar o modificar datos de un origen de datos, ejecute una consulta sobre apodos utilizando las sentencias SELECT, INSERT, UPDATE y DELETE. Las consultas de DB2 SQL se envían a la base de datos federada.

Puede unir datos procedentes de tablas locales y orígenes de datos remotos mediante una sola sentencia de SQL, como si todos los datos fueran locales. Por ejemplo, puede unir datos que residen en los objetos siguientes:

- Una tabla DB2 local contenida en la base de datos federada, una tabla Oracle y una vista Sybase
- Una tabla de DB2 UDB para z/OS situada en un servidor, una tabla de DB2 UDB para z/OS situada en otro servidor y una hoja de cálculo Excel

Mediante el proceso de sentencias de SQL como si los orígenes de datos fueran tablas o vistas relacionales ordinarias dentro de la base de datos federada, el sistema federado puede unir datos relacionales con datos que tienen formatos no relacionales.

Las tablas y vistas de la base de datos federada son *objetos locales*. No cree apodos para estos objetos. En su lugar, utilice el nombre real del objeto en las sentencias SQL.

Los *objetos remotos* son objetos no situados en la base de datos federada. Es necesario crear apodos para estos objetos. Por ejemplo:

- Tablas y vistas situadas en otra base de datos o instancia del sistema federado
- Tablas y vistas situadas en otra base de datos o instancia de otro sistema
- Tablas y vistas situadas en orígenes de datos tales como Oracle, Sybase y ODBC

Cursores declarados WITH HOLD

Puede utilizar la opción WITH HOLD en un cursor que esté definido en un apodo.

Para el derivador DRDA y el origen de datos DB2 Database para Linux, UNIX y Windows, los cursores que se declaran mediante la opción WITH HOLD permanecen abiertos en varias unidades de trabajo.

Si declara un cursor WITH HOLD y el origen de datos no es compatible con esta opción, el atributo no se tiene en cuenta.

Desencadenantes

Un apodo no puede ser un destino de actualización en un desencadenante.

Puede incluir sentencias SELECT para apodos en el cuerpo del desencadenante. No puede incluir sentencias INSERT, UPDATE ni DELETE para apodos en el cuerpo del desencadenante.

Acceso a datos mediante apodos

En un sistema federado puede acceder fácilmente a los datos sin importar dónde estén ubicados. Para acceder a los datos se crean apodos correspondientes a los objetos de origen de datos, tales como tabla y vistas.

Por ejemplo, si el apodo DEPT representa la tabla remota EUROPE.PERSON.DEPT, puede utilizar la sentencia SELECT * FROM DEPT para consultar información de la tabla remota. Cuando realiza una consulta con un apodo, no necesita recordar los detalles de conexión del origen de datos. Por tanto, no necesita conocer lo siguiente:

- El nombre del objeto contenido en el origen de datos
- El servidor donde reside el objeto del origen de datos
- El tipo de origen de datos donde reside el objeto, tal como Informix y Oracle
- El lenguaje de consulta o dialecto de SQL utilizado por el origen de datos
- Las correlaciones de tipos de datos entre el origen de datos y el servidor federado
- Las correlaciones de funciones entre el origen de datos y el servidor federado

Los metadatos asociados al catálogo de la base de datos federada proporcionan al servidor federado la información necesaria para procesar las consultas realizadas por el usuario. Estos metadatos se obtienen de los orígenes de datos cuando el servidor federado y la base de datos se instalan y configuran para acceder a los orígenes de datos.

Una vez instalado el sistema federado, utilice apodos para consultar los orígenes de datos o refinar la configuración del sistema federado.

Sentencias SQL que pueden utilizarse con apodos

Puede utilizar apodos en sentencias de SQL.

Tabla 15. Sentencias de SQL habituales para utilizar con apodos

sentencia de SQL	Descripción	Autorización necesaria
ALTER NICKNAME	Modifica un apodo existente mediante la modificación del nombre de columna local, el tipo de datos local, las opciones de columna federadas o restricciones informativas. La tabla o vista contenida en el origen de datos permanece inalterada.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio ALTER o CONTROL para el apodo • Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe • Definidor del apodo, tal como está registrado en la columna DEFINER de la vista de catálogo del apodo

Tabla 15. Sentencias de SQL habituales para utilizar con apodos (continuación)

sentencia de SQL	Descripción	Autorización necesaria
ALTER TABLE	Cambia una tabla remota que se creó mediante la base de datos federada utilizando DDL transparente. No puede alterar tablas que se crearon de forma nativa en el origen de datos. Puede utilizar restricciones informativas para añadir un restricción de integridad referencial a un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio ALTER o CONTROL para el apodo • Privilegio ALTERIN en el esquema, si el nombre de esquema del apodo existe
COMMENT ON	Añade o sustituye comentarios en las descripciones de catálogo para diversos objetos, tales como funciones, correlaciones de funciones, índices, apodos, servidores, opciones de servidor, correlaciones de tipos y derivadores.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio ALTER o CONTROL para el objeto • Privilegio ALTERIN para el esquema • Definidor del objeto, tal como está registrado en la columna DEFINER de la vista de catálogo del objeto
CREATE ALIAS	Define un alias para un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Autorización IMPLICIT_SCHEMA sobre la base de datos, si el nombre de esquema implícito o explícito del alias no existe • Privilegio CREATEIN sobre el esquema, si el nombre de esquema del alias hace referencia a un esquema existente
CREATE INDEX con la cláusula SPECIFICATION ONLY	Crea una especificación de índice (metadatos) que indica al optimizador de consultas que un objeto de origen de datos tiene un índice. No se crea ningún índice propiamente dicho, solamente se crea la especificación.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio CONTROL o INDEX sobre el objeto de origen de datos asociado — y la autorización IMPLICIT_SCHEMA sobre la base de datos o bien el privilegio CREATEIN sobre el esquema
CREATE TABLE con la cláusula OPTIONS	Crea una tabla remota mediante la base de datos federada utilizando DDL transparente.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio CREATETAB sobre la base de datos y privilegio USE sobre el espacio de tabla — y la autorización IMPLICIT_SCHEMA sobre la base de datos o bien el privilegio CREATEIN sobre el esquema

Tabla 15. Sentencias de SQL habituales para utilizar con apodos (continuación)

sentencia de SQL	Descripción	Autorización necesaria
CREATE TABLE con las cláusulas AS fullselect y DATA INITIALLY DEFERRED REFRESH	Crea una tabla de consultas materializadas utilizando una selección completa donde se especifica un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio CREATETAB sobre la base de datos y privilegio USE sobre el espacio de tabla — y la autorización IMPLICIT_SCHEMA sobre la base de datos o bien el privilegio CREATEIN sobre el esquema • Privilegio CONTROL sobre la tabla o vista • Privilegio SELECT sobre la tabla o vista y privilegio ALTER si se especifica REFRESH DEFERRED
CREATE VIEW	Crea una vista donde se especifican uno o más apodos.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio CONTROL o SELECT sobre el apodo — y la autorización IMPLICIT_SCHEMA sobre la base de datos o bien el privilegio CREATEIN sobre el esquema
DELETE	Suprime filas del objeto de origen de datos, tal como una tabla o vista que tenga un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio DELETE sobre el apodo y privilegio DELETE sobre el objeto de origen de datos asociado • Privilegio CONTROL sobre el objeto de origen de datos asociado
DROP	Suprime un objeto, tal como un apodo, vista federada o especificación de índice. La tabla, vista o índice que reside en el origen de datos permanece inalterado. Cuando se eliminan tablas que se crearon utilizando DDL transparente, también se elimina el apodo correspondiente a la tabla.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio DROPIN sobre el esquema correspondiente al objeto • Privilegio CONTROL sobre el objeto
GRANT	Otorga privilegios sobre apodos y vistas federadas, tales como ALTER, DELETE, INDEX, INSERT, SELECT o UPDATE. Los privilegios del origen de datos se deben otorgar por separado.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • WITH GRANT OPTION para cada privilegio identificado • Privilegio CONTROL sobre el objeto

Tabla 15. Sentencias de SQL habituales para utilizar con apodos (continuación)

sentencia de SQL	Descripción	Autorización necesaria
INSERT	Inserta filas en el objeto de origen de datos, tal como una tabla o vista que tenga un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio INSERT sobre el apodo y privilegio INSERT sobre el objeto de origen de datos asociado • Privilegio CONTROL sobre el objeto de origen de datos asociado
LOCK TABLE	Hace que se bloquee el objeto remoto situado en el origen de datos. Impide que procesos de aplicación simultáneos modifiquen una tabla de origen de datos que tenga un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio SELECT sobre la tabla asociada • Privilegio CONTROL sobre la tabla asociada.
REVOKE	Revoca privilegios sobre apodos y vistas federadas, tales como ALTER, DELETE, INDEX, INSERT, SELECT o UPDATE. Los privilegios del origen de datos se deben revocar por separado.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio CONTROL sobre el objeto
SELECT	Selecciona filas del objeto de origen de datos, tal como una tabla o vista que tenga un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio SELECT sobre el apodo y privilegio SELECT sobre el objeto de origen de datos asociado • Privilegio CONTROL sobre el objeto de origen de datos asociado
UPDATE	Actualiza los valores contenidos en columnas especificadas de filas del objeto de origen de datos, tales como una tabla o vista que tenga un apodo.	<ul style="list-style-type: none"> • Autorización SYSADM o DBADM • Privilegio UPDATE sobre el apodo y privilegio UPDATE sobre el objeto de origen de datos asociado • Privilegio CONTROL sobre el objeto de origen de datos asociado

Cuando envía una consulta a la base de datos federada, se comprueban los privilegios de autorización sobre el apodo contenidos en la consulta. Los requisitos de autorización para el objeto de origen de datos especificado por el apodo solamente se aplican cuando la consulta se procesa en el origen de datos.

Para seleccionar, insertar, actualizar o suprimir datos mediante un apodo, el ID de autorización de la sentencia debe incluir estos privilegios:

- El privilegio apropiado sobre el apodo para que la base de datos federada acepte la petición

- El privilegio apropiado sobre el objeto de tabla asociado para que el origen de datos acepte la petición

Por ejemplo, para actualizar un origen de datos mediante un apodo, es necesario el privilegio UPDATE sobre el apodo y el privilegio UPDATE sobre el objeto de origen de datos asociado.

Acceso a nuevos objetos de origen de datos

Para acceder a nuevos objeto de origen de datos, debe crear apodos para ellos. Utilice la sentencia CREATE NICKNAME para los orígenes de datos que no tengan apodos.

Antes de empezar

El sistema federado debe estar configurado para acceder al origen de datos.

La base de datos federada debe contener una definición de servidor para el servidor de origen de datos donde reside el objeto. Utilice la sentencia CREATE SERVER para crear una definición de servidor.

Para insertar, actualizar o suprimir datos utilizando un apodo, se deben cumplir estas condiciones:

- Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir los privilegios necesarios SELECT, INSERT, UPDATE y DELETE sobre el apodo para que la base de datos federada acepte la petición.
- El ID de usuario del origen de datos debe tener los privilegios necesarios SELECT, INSERT, UPDATE y DELETE sobre el objeto de tabla asociado para que el origen de datos acepte la petición.
- El ID de usuario del origen de datos se debe correlacionar con el ID de autorización del servidor federado mediante una correlación de usuarios.

El usuario debe tener las autorizaciones necesarias para la sentencia CREATE NICKNAME.

- Autorización SYSADM o DBADM
- Autorización IMPLICIT_SCHEMA sobre la base de datos federada, si el nombre de esquema implícito o explícito del apodo no existe
- Privilegio CREATEIN para el esquema, si el nombre de esquema del apodo existe

Acerca de esta tarea

Periódicamente necesita acceder a objetos de origen de datos que carecen de apodos. Estos objetos pueden ser nuevos objetos añadidos a un origen de datos, tales como una vista recién creada. Pueden también ser objetos existentes que no se registraron en el servidor federado cuando se configuró inicialmente. En cualquiera de los dos casos, debe crear un apodo para el objeto.

Procedimiento

Para acceder a nuevos objetos de origen de datos, emita la sentencia CREATE NICKNAME.

Creación de apodos para orígenes de datos relacionales y no relacionales

La sentencia CREATE NICKNAME difiere ligeramente según se utilice para orígenes de datos relacionales o no relacionales.

Para orígenes de datos relacionales, la sintaxis de la sentencia CREATE NICKNAME es:

```
CREATE NICKNAME nombre_apodo FOR nombre_servidor."esquema_remoto"."nombre_objeto"
```

nombre_apodo

Apodo exclusivo para designar el objeto de origen de datos. Los apodos pueden tener una longitud de hasta 128 caracteres.

El nombre de apodo consta de dos partes: el esquema y el apodo. Si omite el esquema al crear el apodo, el esquema del apodo será el ID de autenticación del usuario creador del apodo. Los valores por omisión del nombre de esquema se seleccionan de acuerdo con la jerarquía siguiente:

1. Registro especial CURRENT SCHEMA
2. Registro especial SESSION_USER
3. Registro especial SYSTEM USER
4. ID de autorización conectado a la base de datos

```
FOR nombre_servidor."esquema_remoto"."nombre_objeto"
```

Identificador formado por tres partes que se utiliza para designar el objeto de origen de datos remota. Si el origen de datos no es compatible con la utilización de esquemas, omita el esquema en la sentencia CREATE NICKNAME.

- *nombre_servidor* es el nombre asignado al servidor del origen de datos en la sentencia CREATE SERVER.
- *esquema_remoto* es el nombre del esquema remoto al que pertenece el objeto.
- *nombre_objeto* es el nombre del objeto remoto al que se desea acceder.

OPTIONS (*lista_opciones*)

Información sobre el apodo que permite al compilador de consultas de SQL y al derivador ejecutar consultas de forma eficiente.

Para algunos orígenes de datos no relacionales, la sintaxis de la sentencia CREATE NICKNAME es:

```
CREATE NICKNAME nombre_apodo lista_definiciones_columna  
FOR SERVER nombre_servidor  
OPTIONS (lista_opciones)
```

nombre_apodo

Apodo exclusivo para designar el objeto de origen de datos, tal como se describió anteriormente para los orígenes de datos relacionales.

lista_definiciones_columna

Lista de columnas y tipos de datos de apodos.

FOR SERVER *nombre_servidor*

Nombre local creado para el servidor remoto en la definición de servidor de la sentencia CREATE SERVER.

OPTIONS (*lista_opciones*)

Información sobre el apodo que permite al compilador de consultas de SQL y al derivador ejecutar consultas de forma eficiente.

Nombres de índice y de columnas de apodos

Cuando se crea un apodo, el servidor federado utiliza un algoritmo para generar nombres de índice y de columnas de apodos.

Este algoritmo se mejora en la versión 9.5 para poder establecer una correspondencia lo más precisa posible entre los nombres de índice y de columnas de apodo y los nombres originales.

El algoritmo sólo se aplica a orígenes de datos relacionales. En el caso de los orígenes de datos no relacionales, el nombre especificado en la sentencia CREATE NICKNAME no sufre ningún cambio.

En la tabla siguiente se muestran ejemplos de nombres de columnas de apodos derivados de nombres de columnas remotas. En estos ejemplos, las columnas remotas se encuentran en tablas remotas distintas. Los nombres de columnas de apodos que son distintos del nombre de columna remota se indican en negrita.

Tabla 16. Nombre de columnas de apodos generados a partir de nombres de columnas remotas

Nombre de columna remota	Nombre resultantes del algoritmo actual para los orígenes de datos que...			Nombre resultante antes de la versión 9.5 (el algoritmo convierte los caracteres no alfanuméricos)
	Convertir identificadores a mayúsculas	Convertir identificadores a minúsculas	No modificar los identificadores	
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 las columnas remotas se encuentran en la misma tabla, el servidor federado utiliza el algoritmo existente que genera nombres exclusivos. Por ejemplo, en el caso de los nombres de columnas de apodos de los orígenes de datos que no modifican los identificadores, los nombres de las columnas remotas dan lugar a estos nombres:

- column1 da lugar a COLUMN1
- Column1 da lugar a COLUMN10
- COLUMN1 da lugar a COLUMN11

Acceso a orígenes de datos mediante sesiones de paso a través

Puede enviar sentencias de SQL directamente a los orígenes de datos utilizando una modalidad especial denominada paso a través. La sesión de paso a través le permite enviar sentencias de SQL en el dialecto de SQL correspondiente a ese origen de datos.

Utilice una sesión de paso a través cuando desee realizar una operación que no sea posible con SQL. Por ejemplo, utilice una sesión de paso a través para crear un procedimiento, para crear un índice o para realizar consultas en el dialecto nativo del origen de datos.

Los orígenes de datos que se pueden utilizar con la modalidad de paso a través solamente aceptan sentencias de SQL dentro de una sesión de paso a través.

De forma similar, puede utilizar una sesión de paso a través para realizar acciones que no están soportadas por SQL, tales como determinadas tareas administrativas. Las tareas administrativas que puede ejecutar dependen del origen de datos. Por ejemplo, para DB2 Database para Linux, UNIX y Windows, puede ejecutar el programa de utilidad de estadísticas para el origen de datos, pero no puede iniciar ni detener la base de datos remota.

Puede consultar un solo origen de datos cada vez en una sesión de paso a través. Utilice el mandato SET PASSTHRU para abrir una sesión. Utilice el mandato SET PASSTHRU RESET para cerrar la sesión de paso a través. Si utiliza el mandato SET PASSTHRU en lugar del mandato SET PASSTHRU RESET, se cierra la sesión de paso a través actual y se abre una nueva.

En una sesión de paso a través puede declarar un cursor WITH HOLD. Si utiliza esta opción con el derivador DRDA y el origen de datos DB2 Database para Linux, UNIX y Windows, los cursores que declare permanecerán abiertos en varias unidades de trabajo. Si declara un cursor WITH HOLD y el origen de datos no es compatible con esta opción, el atributo no se tiene en cuenta.

No puede utilizar sesiones de paso a través con orígenes de datos no relacionales.

Acceso a datos heterogéneos mediante vistas federadas

Una *vista federada* es una vista de la base de datos federada cuyas tablas base están situadas en orígenes de datos remotos. La vista federada hace referencia a las tablas base utilizando apodos, en lugar de utilizar los nombres de tablas del origen de datos.

Antes de empezar

Para emitir la sentencia CREATE VIEW es necesaria una de las autorizaciones siguientes:

- Autorización SYSADM o DBADM
- Para cada apodo contenido en una selección completa, estas dos autorizaciones:
 - Privilegio CONTROL o SELECT sobre la tabla o vista asociada
 - Una de las autorizaciones o privilegios siguientes:
 - Autorización IMPLICIT_SCHEMA sobre la base de datos federada, si el nombre de esquema implícito o explícito de la vista no existe
 - Privilegio CREATEIN sobre el esquema, si el nombre de esquema de la vista hace referencia a un esquema existente

Los privilegios sobre los objetos asociados no se tienen en cuenta al definir una vista para un apodo de base de datos federada.

Restricciones

Las vistas federadas que se han creado con sentencias UNION ALL son vistas de sólo lectura.

Las vistas federadas que incluyen más de un apodo en la cláusula FROM son vistas de sólo lectura.

Las vistas federadas que incluyen un solo apodo en la cláusula FROM pueden ser vistas de sólo lectura.

- Si el apodo contenido en la cláusula FROM es para un origen de datos no relacional, la vista federada es de sólo lectura.
- Si incluye otros apodos como predicados o subconsultas al crear la vista, la vista federada se puede actualizar.

Acerca de esta tarea

Cuando realiza una consulta desde una vista federada, los datos se obtienen del origen de datos remoto. La acción de crear una vista de base de datos federada a partir de los datos de un origen de datos se denomina a veces “creación de una vista para un apodo”. Esto es debido a que se especifican apodos en lugar de orígenes de datos cuando se crea la vista.

Estas vistas proporcionan un alto grado de independencia respecto de los datos para una base de datos integrada globalmente, de la misma manera que lo hacen las vistas definidas sobre varias tablas locales para gestores centralizados de bases de datos relacionales.

Procedimiento

Para crear una vista federada, emita la sentencia CREATE VIEW.

Cuando se procesa la consulta, se aplican requisitos de autorización del origen de datos para la tabla o vista referenciada por el apodo. El ID de autorización de la sentencia se puede correlacionar con un ID de autorización remoto diferente mediante una correlación de usuarios.

Creación de vistas federadas - ejemplos

Estos ejemplos muestran cómo crear vistas federadas para acceder a datos procedentes de varios orígenes de datos. Los ejemplos muestran la sintaxis de la sentencia CREATE VIEW para la federación.

Ejemplo: creación de una vista federada que fusiona datos similares procedentes de varios objetos de origen de datos

Suponga que trabaja con datos de cliente situados en servidores separados: uno en Europa, uno en Asia y uno en Sudamérica. Los datos de cliente de Europa están en una tabla Oracle. El apodo para dicha tabla es ORA_EU_CUST. Los datos de cliente de Asia están en una tabla Sybase. El apodo para dicha tabla es SYB_AS_CUST. Los datos de cliente de Sudamérica están en una tabla Informix. El apodo para dicha tabla es INFMX_SA_CUST. Cada tabla tiene columnas que contienen el número del cliente (CUST_NO), el nombre del cliente (CUST_NAME), el número del producto (PROD_NO) y la cantidad pedida (QUANTITY). Para crear una vista a partir de estos apodos que fusione estos datos de cliente, emita esta sentencia:

```
CREATE VIEW FV1
AS SELECT * FROM ORA_EU_CUST
UNION
SELECT * FROM SYB_AS_CUST
UNION
SELECT * FROM INFMX_SA_CUST
```

Ejemplo: unión de datos para crear una vista federada

Se está trabajando con datos de cliente en un servidor y datos de ventas en otro servidor. Los datos de cliente están en una tabla Oracle. El apodo para dicha tabla es ORA_EU_CUST. Los datos de ventas están en una tabla de Sybase. El apodo para dicha tabla es SYB_AS_SALES. Desea asociar la información de cliente con las compras realizadas por los clientes. Cada tabla tiene una columna que contiene el número del cliente (CUST_NO). Para crear una vista federada a partir de estos apodos que combine estos datos, emita esta sentencia:

```
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
```

Creación de un apodo para un apodo

Puede crear un apodo para un apodo.

Procedimiento

Siga los pasos de este ejemplo para crear un apodo para un apodo.

Ejemplo: Acceso a una hoja de cálculo Microsoft Excel de un servidor federado AIX

Dispone de un servidor federado en AIX y un servidor federado en Windows. Desea acceder a una hoja de cálculo Excel de ambos servidores federados. Pero el derivador Excel solamente se puede utilizar con servidores federados en Windows. Para acceder a la hoja de cálculo Excel del servidor federado AIX:

1. En el servidor federado Windows, instale IBM InfoSphere Federation Server.
2. Configure el servidor federado Windows para acceder a orígenes de datos Excel.
3. En el servidor federado Windows, cree un apodo para la hoja de cálculo Excel.
4. En el servidor federado AIX, instale IBM InfoSphere Federation Server.
5. Configure el servidor federado AIX para acceder a orígenes de datos DB2.
6. En el servidor federado AIX, cree un apodo para el apodo Excel en el servidor federado Windows.

Selección de datos en un sistema federado

La forma de seleccionar datos en un sistema federado depende del tipo de origen de datos desde donde realice la selección.

Algunos de los tipos de peticiones distribuidas utilizadas en un sistema federado son peticiones que consultan:

- Un origen de datos remoto individual
- Un origen de datos local y un origen de datos remoto
- Varios orígenes de datos remotos

- Una combinación de orígenes de datos remotos y locales

Para seleccionar datos de los orígenes de datos, utilice los apodos de los objetos de origen de datos en la sentencia SELECT.

La base de datos federada es un origen de datos local. Las tablas y vistas de la base de datos federada son objetos locales. Para estos objetos no se crean apodos, sino que se utiliza el nombre real del objeto en las sentencias SELECT.

Los orígenes de datos remotos pueden ser: otra instancia de base de datos DB2 Database para Linux, UNIX y Windows en el servidor federado, otra instancia de base de datos DB2 Database para Linux, UNIX y Windows en otro servidor, y otros orígenes de datos. Los objetos que residen en orígenes de datos remotos son objetos remotos.

Selección de datos en un sistema federado - ejemplos

Este tema muestra un caso en el que un servidor federado accede a varios orígenes de datos y proporciona ejemplos de sentencias SELECT.

Ejemplo: un servidor federado está configurado para acceder a un origen de datos de DB2 para z/OS, un origen de datos de DB2 para System i y un origen de datos de Oracle. Cada origen de datos contiene una tabla donde reside la información sobre ventas. Esta configuración está representada en la figura siguiente.

Clientes de DB2 (aplicación y usuario final)

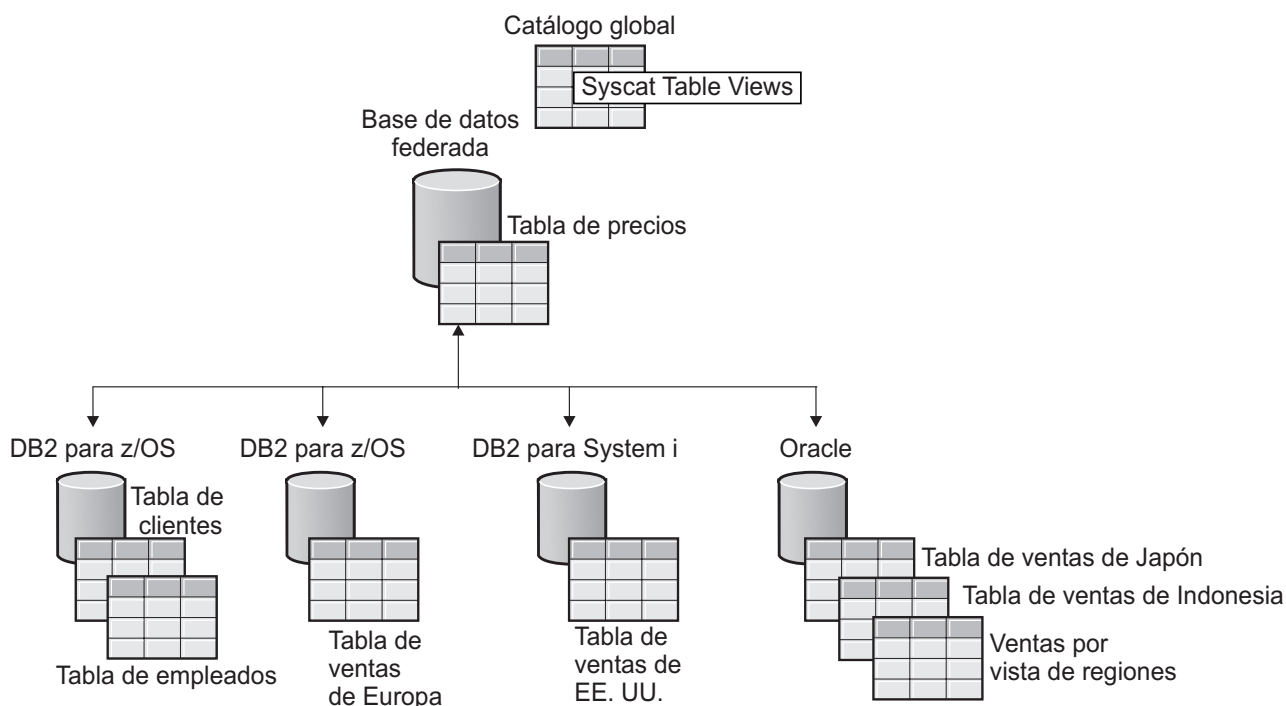


Figura 8. Sistema federado de ejemplo con orígenes de datos DB2 y Oracle

Las tablas de ventas contienen columnas donde se registra el número de cliente (CUST_NO), el volumen del pedido (QUANTITY) y el número de producto solicitado (PROD_NO). Existe también una tabla local en la base de datos federada

que contiene información sobre precios. La tabla de precios contiene columnas donde se registran el número de producto (PROD_NO) y el precio actual (PRICE).

Los apodos para los objetos del origen de datos remoto están contenidos en las tablas SYSCAT.TABLES, tal como muestra en las tablas siguientes. La columna TYPE indica el tipo de objeto, tal como apodo (A), tabla local (T) o vista (V).

Tabla 17. Información del origen de datos

Nombre de objeto de origen de datos	Tipo de objeto	Ubicación
PRICES	Tabla local	Base de datos federada
EUROPE_SALES	Tabla remota	Base de datos DB2 para z/OS
US_SALES	Tabla remota	Base de datos DB2 para System i
JAPAN_SALES	Tabla remota	Base de datos Oracle
SALES_BY_REGION	Vista remota	Base de datos Oracle

Tabla 18. Tablas SYSCAT

TABNAME	TYPE
PRICES	T
FED_PRICES	N
Z_EU_SALES	N
Si_US_SALES	N
ORA_JAPANSALES	N
ORA_REGIONSALES	N
...	

Para seleccionar datos utilizando un apodo, se deben cumplir estas condiciones:

- Los privilegios pertenecientes al ID de autorización de la sentencia deben incluir el privilegio SELECT para el apodo (para que la base de datos federada acepte la petición).
- El ID de usuario contenido en el origen de datos debe tener el privilegio SELECT para el objeto de tabla asociado (para que el origen de datos acepte la petición).
- El ID de usuario del origen de datos se debe correlacionar con el ID de autorización del servidor federado mediante una correlación de usuarios.

Los ejemplos siguientes de sentencias SELECT utilizan el sistema federado de ejemplo descrito anteriormente.

Ejemplo: consulta de un origen de datos individual

Z_EU_SALES contiene los productos ordenados de acuerdo con los clientes de Europa. También incluye el volumen de cada venta. Esta consulta utiliza una sentencia SELECT con una cláusula ORDER BY para listar las ventas de Europa y clasifica la lista por el número de cliente:

```
SELECT CUST_NO, PROD_NO, QUANTITY
       FROM Z_EU_SALES
       ORDER BY CUST_NO
```

Ejemplo: unión de un origen de datos local y un origen de datos remoto

PRICES es una tabla que reside en la base de datos federada. Contiene la lista de precios de los productos en venta. Este ejemplo selecciona los precios de esa tabla local que corresponden a los productos listados en Z_EU_SALES. El conjunto resultante se clasifica por el número de cliente.

```
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
```

Ejemplo: consulta de varios orígenes de datos remotos

Este ejemplo recoge toda la información de ventas de cada región y clasifica el resultado por el número de producto.

```
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
```

Una vista del origen de datos Oracle muestra las ventas correspondientes a Japón e Indonesia. El apodo de esta vista es ORA_SALESREGION. Esta información de ventas se combina con las ventas de Estados Unidos y junto a cada venta se muestra el precio de los productos.

```
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
```

Restricciones informativas para apodos

Puede utilizar restricciones informativas para apodos a fin de mejorar el rendimiento de las consultas. Las restricciones informativas son reglas que el optimizador utiliza para mejorar el rendimiento, pero que el gestor de bases de datos no aplica.

Puede especificar lo siguiente para apodos:

- Restricciones referenciales
- Restricciones de comprobación
- Restricciones de dependencia funcional
- Restricciones de clave primaria
- Restricciones de unicidad

Especificación de restricciones informativas para apodos (Centro de control de DB2)

Para mejorar el rendimiento de las consultas en orígenes de datos remotos, puede especificar restricciones informativas para apodos. Pero el servidor federado no aplica ni comprueba las restricciones. Puede especificar restricciones informativas para un apodo desde el Centro de control de DB2 o la línea de mandatos de DB2.

Restricciones

Después de definir restricciones informativas para un apodo, solamente puede modificar los nombres de columnas para ese apodo después de eliminar las restricciones informativas.

Acerca de esta tarea

Para los orígenes de datos relacionales, puede especificar restricciones informativas cuando modifica un apodo.

Para los orígenes de datos no relacionales, puede especificar restricciones informativas cuando crea o modifica un apodo.

Procedimiento

Para especificar restricciones informativas para apodos desde el Centro de control de DB2:

1. Seleccione la carpeta **Apodos**:
 - Si está creando un apodo, en el panel de detalles del objeto del Centro de control de DB2, pulse **Crear apodos** en la lista de acciones. Se abrirá el asistente para Apodos.
 - Desde la ventana Crear apodos, abra el cuaderno Añadir apodo o el cuaderno Propiedades correspondiente a un apodo:
 - a. Si piensa crear un apodo individual, pulse **Añadir**. Se abrirá el cuaderno Añadir apodo.
 - b. Si la función de descubrimiento creó una lista de apodos, seleccione el apodo al que desee añadir restricciones informativas. Luego pulse **Propiedades**. Se abrirá el cuaderno Propiedades.
 - Si está modificando un apodo, pulse en el apodo que desee modificar. En el panel de detalles del objeto del Centro de control de DB2, pulse **Alterar** en la lista de acciones. Se abrirá el cuaderno Modificar apodo.
2. En la página Claves, defina las restricciones de integridad referencial para el apodo. Puede definir una restricción de clave primaria, de clave exclusiva o de clave foránea.
3. En la página Restricciones de comprobación, defina las restricciones de comprobación o de dependencia funcional para el apodo.
4. Pulse **Aceptar** para definir las restricciones informativas y cerrar el cuaderno.

Especificación de restricciones informativas para apodos (línea de mandatos de DB2)

Para mejorar el rendimiento de las consultas en orígenes de datos remotos, puede añadir restricciones informativas para apodos. Pero el servidor federado no aplica ni comprueba las restricciones. Puede especificar restricciones informativas para un apodo desde el Centro de control de DB2 o la línea de mandatos de DB2.

Restricciones

Después de definir restricciones informativas para un apodo, solamente puede modificar los nombres de columnas para ese apodo después de eliminar las restricciones informativas.

Acerca de esta tarea

Para los orígenes de datos relacionales, puede especificar restricciones informativas cuando modifica un apodo.

Para los orígenes de datos no relacionales, puede especificar restricciones informativas cuando crea o modifica un apodo.

Procedimiento

Para especificar restricciones informativas para un apodo desde la línea de mandatos de DB2, emita la sentencia CREATE NICKNAME o la sentencia ALTER NICKNAME con los atributos de restricción apropiados.

Especificación de restricciones informativas para apodos - ejemplos

Estos ejemplos muestran el uso de restricciones informativas para apodos. Utilice las sentencias CREATE o ALTER NICKNAME para definir restricciones de comprobación, restricciones referenciales y otras estructuras de datos.

Ejemplo: restricción de comprobación informativa

En la tabla remota siguiente, los datos de la columna "salary" son siempre mayores que 10000.

```
CREATE TABLE account.salary (  
    empno INTEGER NOT NULL PRIMARY KEY,  
    salary INTEGER NOT NULL  
);
```

Cree un apodo para esta tabla:

```
CREATE NICKNAME account.salary FOR myserv.account.salary;
```

A continuación, añada restricciones de comprobación informativas para el apodo emitiendo esta sentencia:

```
ALTER NICKNAME account.salary ADD CONSTRAINT cons1 CHECK( salary > 10000 )  
NOT ENFORCED  
ENABLE QUERY OPTIMIZATION;
```

Ejemplo: restricción referencial informativa entre apodos

Este ejemplo utiliza dos apodos, N1 y N2. La columna F1 del apodo N2 contiene el valor de clave de la columna P1 del apodo N1. Puede definir restricciones referenciales para el apodo N2 emitiendo esta sentencia:

```
ALTER NICKNAME SCHEMA1.N2 ADD CONSTRAINT ref1  
    FOREIGN KEY (F1) REFERENCES SCHEMA1.N1 (P1)  
    NOT ENFORCED;
```

Ejemplo: restricción referencial informativa entre un apodo y una tabla

En este ejemplo, la columna F1 del apodo N3 contiene el valor de clave de la columna P1 de la tabla T1. Puede definir restricciones referenciales para el apodo N3 emitiendo esta sentencia:

```
ALTER NICKNAME SCHEMA1.N3 ADD CONSTRAINT ref1
    FOREIGN KEY (F1) REFERENCES SCHEMA1.T1 (P1)
    NOT ENFORCED;
```

Ejemplo: restricción referencial informativa entre una tabla y un apodo

En este ejemplo, la columna F1 de la tabla T2 contiene el valor de clave de la columna P1 del apodo N4. Puede definir restricciones referenciales para la tabla T2 emitiendo esta sentencia:

```
ALTER TABLE SCHEMA1.T2 ADD CONSTRAINT ref1
    FOREIGN KEY (F1) REFERENCES SCHEMA1.N4 (P1)
    NOT ENFORCED;
```

Ejemplo: dependencia funcional

En este ejemplo, el par de columnas C1 y C2 determinan de forma unívoca el valor de la columna P1. Puede definir la dependencia funcional emitiendo esta sentencia:

```
ALTER NICKNAME SCHEMA1.NICK1 ADD CONSTRAINT FD1 CHECK( P1 DETERMINED BY (C1,C2) )
    NOT ENFORCED ENABLE QUERY OPTIMIZATION;
```

Ejemplo: archivo con estructura de tabla

Esta sentencia define una clave primaria para un archivo con estructura de tabla:

```
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');
```

Esquema en estrella

Esta sentencia crea cuatro tablas de dimensiones y una tabla de datos:

```
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,
    ...
);

```

El servidor federado crea los apodos siguientes para la tabla de datos y las tablas de dimensiones:

```

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;

```

Puede definir la siguiente relación de clave foránea emitiendo estas sentencias:

```

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;

```

Cuando valor de la columna TEL del apodo CUSTOMER es exclusivo, puede añadir la siguiente restricción de unicidad informativa con esta sentencia:

```

ALTER NICKNAME SCHEMA.CUSTOMER ADD CONSTRAINT U1 UNIQUE( TEL )
    NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Cuando el valor de la columna SHOP_ID del apodo LOCATION determina de forma unívoca el valor de la columna LOCATION_ID, puede definir la siguiente dependencia funcional con esta sentencia:

```

ALTER NICKNAME SCHEMA.LOCATION
ADD CONSTRAINT F1 CHECK( LOCATION_ID DETERMINED BY SHOP_ID )
    NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Debido a que el valor de la columna QUARTER del apodo TIMEDIM está comprendido entre 1 y 4, puede definir la siguiente restricción de comprobación informativa con esta sentencia:

```

ALTER NICKNAME SCHEMA.TIMEDIM
ADD CONSTRAINT Q1 CHECK( QUARTER BETWEEN 1 AND 4 )
    NOT ENFORCED ENABLE QUERY OPTIMIZATION;

```

Las sentencias de este ejemplo crean apodos para tablas remotas. Los apodos tienen claves primarias cuando las tablas remotas tienen claves primarias. Cuando crea apodos sobre vistas, los apodos carecen de claves primarias. En este caso, puede cambiar el apodo para añadir una restricción informativa de clave primaria. Por ejemplo:

```
CREATE NICKNAME SCHEMA.LOCATION FOR SERVER.SH.V_LOCATION;  
ALTER NICKNAME SCHEMA.LOCATION  
ADD CONSTRAINT P1 PRIMARY KEY ( LOCATION_CODE ) NOT ENFORCED;
```

Actualización de estadísticas para apodos

La recuperación de estadísticas para un apodo garantiza que el optimizador de consultas utiliza la información más reciente sobre el apodo cuando genera planes de acceso a consultas.

Recurso de actualización de estadísticas de apodo - Visión general

Puede obtener estadísticas para un apodo a fin de proporcionar al optimizador de consultas información exacta y actual sobre el apodo. El optimizador utiliza esta información para determinar un plan de acceso óptimo para una consulta.

Cuando el usuario registra un apodo para un objeto del origen de datos, el servidor federado añade información sobre el objeto al catálogo del sistema situado en la base de datos federada. El optimizador de consultas utiliza esta información para planificar la obtención de datos a partir del objeto del origen de datos. La base de datos federada no detecta automáticamente los cambios hechos en los objetos del origen de datos, por lo que la información del catálogo del sistema puede ser obsoleta.

Puede obtener las estadísticas disponibles actualmente sobre apodos, columnas e índices de la base de datos a partir del origen de datos remoto. Para recuperar y actualizar estadísticas de apodos, utilice el Centro de control de DB2 o el procesador de línea de mandatos de DB2.

Generalmente, puede utilizar el Centro de control de DB2 o el procesador de línea de mandatos de DB2 para actualizar manualmente las estadísticas de apodos cuando se recogen nuevas estadísticas en una tabla remota. Además, las características de recogida automática de estadísticas mantienen las estadísticas actualizadas por omisión.

Generalmente, se recomienda actualizar manualmente las estadísticas de apodos, mediante el Centro de control de DB2 o el procesador de línea de mandatos de DB2, porque la tabla remota sólo recoge las nuevas estadísticas. Además, las características de recogida automática de estadísticas mantienen las estadísticas actualizadas por omisión.

Puede obtener estadísticas sobre apodos para los orígenes de datos relacionales siguientes:

- Familia de productos DB2(DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Puede obtener estadísticas sobre apodos para los orígenes de datos no relacionales siguientes:

- BioRS
- Excel
- Archivos con estructura de tabla
- XML en el apodo raíz

Las estadísticas recogidas para cada origen de datos varían.

Puede actualizar las estadísticas siguientes utilizando el método de recogida de estadísticas basado en catálogos:

- CARD
- FPAGES
- NPAGES
- OVERFLOW
- COLCARD
- HIGH2KEY
- LOW2KEY
- NLEAF
- NLEVELS
- CLUSTERFACTOR
- CLUSTERRATIO
- FULLKEYCARD
- FIRSTKEYCARD

Puede actualizar las estadísticas siguientes utilizando el método de recogida de estadísticas basado en datos:

- CARD
- COLCARD
- HIGH2KEY
- LOW2KEY
- FULLKEYCARD
- FIRSTKEYCARD

Puede obtener estadísticas de un apodo individual o de todos los apodos de un esquema para una definición de servidor determinada. Puede también obtener estadísticas para apodos en objetos origen con Seguridad de etiqueta de Oracle. Si la base de datos está restringida, solamente los usuarios con el acceso apropiado pueden ver las estadísticas HIGH2KEY y LOW2KEY. Para las bases de datos no restringidas, las estadísticas sobre apodos en objetos con Seguridad de etiqueta de Oracle están expuestas y pueden suponer un riesgo de seguridad. En estos casos, puede restringir el acceso a los catálogos del sistema que contengan información confidencial. Si falla cualquier parte de la obtención de datos, la base de datos cancela los cambios.

Puede obtener estadísticas sobre apodos en cualquier sistema operativo que sea compatible con el servidor federado.

Métodos de obtención de estadísticas de apodos

Puede seleccionar el método utilizado para recoger estadísticas y puede limitar sus elecciones a determinadas columnas e índices. El método basado en catálogos tiene un mejor rendimiento. En cambio, el método basado en datos proporciona estadísticas más actuales, pero exige más tiempo para ejecutarse.

Puede seleccionar uno de los métodos siguientes para la recogida de estadísticas:

- Método basado en catálogos

El método basado en catálogos copia estadísticas de tablas de catálogo del origen de datos en la tabla de catálogo federada. Solamente se copian las estadísticas que se pueden correlacionar semánticamente con estadísticas federadas. Sin embargo, las estadísticas de apodos solamente son tan exactas y actuales como lo sea la información contenida actualmente en el catálogo del origen remoto. Si la información estadística es obsoleta, también lo serán las estadísticas de apodos recogidas. Cuando utilice el método basado en catálogos, compruebe que las estadísticas del origen remoto sean actuales.

Debido a que las estadísticas se copian del catálogo del origen remoto al catálogo del servidor federado, normalmente el método basado en catálogos para la recogida de estadísticas es muy rápido.

- Método basado en datos

El método basado en datos no depende de las estadísticas contenidas en el origen remoto. Este método crea sus propias estadísticas empíricamente a partir de los resultados de las consultas que el método emite para los apodos. Las estadísticas recogidas con este método son un reflejo exacto de los datos remotos.

El método basado en datos puede ser lento si el tamaño de fila de los apodos que intervienen es grande. Normalmente las consultas suponen clasificaciones y agregados de gran volumen. Por esta razón, utilice la recogida de estadísticas basada en datos solamente si no se pueden obtener estadísticas satisfactorias con el método basado en catálogos.

Si desea aumentar el rendimiento del método basado en datos a expensas de la calidad de las estadísticas recogidas, limite la recogida de estadísticas a los tipos de columnas e índices para los que el beneficio sea el mayor. Esos tipos de columnas incluyen las columnas que intervienen en predicados, claves de unión, operaciones de agrupamiento, o las columnas que forman parte de uno o más índices.

Cuando se utiliza el método basado en catálogos, normalmente no es necesario limitar la recogida de estadísticas a determinadas columnas o índices, pues la actividad general desplegada por este método de recogida es muy baja.

Obtención de estadísticas de apodo

Puede obtener estadísticas sobre un apodo para que el optimizador de consultas utilice la información sobre el apodo que está disponible en el origen de datos. Puede actualizar las estadísticas de apodo para un apodo individual, varios apodos o todos los apodos.

Acerca de esta tarea

El optimizador de consultas recoge estadísticas, tales como las estadísticas de apodo HIGH2KEY y LOW2KEY, para objetos de origen de datos que hacen uso del control de accesos basado en etiquetas o la seguridad de etiquetas de Oracle. Para las bases de datos que utilizan estos tipos de seguridad, solamente pueden ver las

estadísticas los usuarios con el nivel de acceso apropiado. Para las bases de datos no restringidas, puede restringir el acceso o asignar a las estadísticas de apodo HIGH2KEY y LOW2KEY un valor vacío o carente de significado.

Antes de empezar

Son aplicables los requisitos siguientes cuando actualiza estadísticas desde la línea de mandatos:

- El servidor federado crea el archivo de registro en el servidor. Los directorios especificados en la vía de acceso deben existir.
- El ID de usuario protegido de la instancia federada debe tener autorización para crear el archivo de registro en la ubicación especificada.

Restricciones

El ID de usuario utilizado para conectar con la base de datos federada debe estar correlacionado con la tabla del origen de datos remoto.

Si el nombre o tipo de la columna local no representa la correlación de tipos por omisión con el nombre o tipo de la columna remota, el programa de actualización de estadísticas de apodo no obtiene estadísticas de columna.

Acerca de esta tarea

Por omisión, la recogida de estadísticas se intenta para todas las columnas e índices de un apodo. Puede limitar la recogida de estadísticas a determinadas columnas e índices y especificar un archivo de registro.

Procedimiento

Para actualizar estadísticas de apodo desde la línea de mandatos de DB2 o desde dentro de una aplicación, invoque el procedimiento almacenado SYSPROC.NNSTAT.

Para actualizar estadísticas de apodo desde el Centro de control de DB2:

1. Seleccione los apodos para los que desee obtener estadísticas actuales.
2. Si no existe un catálogo de herramientas de DB2, se muestra una ventana desde la que puede crear el catálogo de herramientas de DB2. Cree el catálogo de herramientas cuando desee planificar la actualización de las estadísticas de apodo.
3. Especifique los valores para la actualización.

Obtención de estadísticas para varios apodos (Centro de control de DB2)

Puede actualizar estadísticas para varios apodos desde el Centro de control de DB2 o la línea de mandatos de DB2.

Procedimiento

Para actualizar estadísticas de apodo para varios apodos desde el Centro de control de DB2:

1. Seleccione los apodos necesarios:
 - Apodos con definición de servidor:

- a. Expanda la carpeta **Objetos de base de datos federada**. Seleccione la carpeta de derivador con la que desee trabajar.
 - b. Expanda la carpeta **Definiciones de servidor**. Seleccione la carpeta de servidor que contenga los apodos con los que desee trabajar.
 - c. Haga una doble pulsación en la carpeta **Apodos**.
 - d. Pulse con el botón derecho del ratón en los nombres de los apodos y seleccione **Estadísticas**.
 - e. Seleccione **Actualizar**. Se abrirá el cuaderno Actualización de estadísticas correspondiente a varios apodos.
 - Para actualizar las estadísticas para varios apodos que está asociados a una determinada definición de base de datos:
 - a. Expanda la carpeta **Bases de datos**. Seleccione la carpeta de la **Base de datos** que contenga las definiciones de apodo con las que desee trabajar.
 - b. Haga una doble pulsación en la carpeta **Apodos**.
 - c. Pulse con el botón derecho del ratón en los nombres de los apodos que desee actualizar y seleccione **Estadísticas**.
 - d. Seleccione **Actualizar**. Se abrirá el cuaderno Actualización de estadísticas correspondiente a varios apodos.
2. Especifique los valores para la actualización:
- Página Actualización de estadísticas:
 - a. Examine los apodos mostrados en la ventana Actualización de estadísticas para comprobar que sean los apodos para los que desea actualizar estadísticas. Puede deselectionar los apodos que no desee actualizar.
 - b. Seleccione **Detalles** para seleccionar estadísticas a nivel de columna y a nivel de índice para actualizar. Se abrirá el cuaderno Detalles.
 - Página Detalles:
 - a. Seleccione todas las columnas del apodo para actualizar o seleccione columnas determinadas para actualizar.
 - b. Seleccione todos los índices del apodo para actualizar o seleccione índices determinados para actualizar.
 - c. Seleccione un archivo de registro existente o escriba la vía de acceso totalmente calificada de un nuevo archivo de registro.
 - Página Método. Seleccione uno de los métodos siguientes para la recogida de estadísticas:
 - a. Método basado en catálogos: válido para apodos relacionales.
 - b. Método basado en datos: válido para apodos relacionales y no relacionales
 - c. Ambos métodos: la opción por omisión
 - **Opcional:** en la página Planificación, especifique cuándo desea que se ejecute la actualización de estadísticas del apodo.

Obtención de estadísticas para un único apodo (Centro de control de DB2)

Puede obtener estadísticas para un apodo individual asociado a una definición de servidor. Puede actualizar estadísticas sobre apodos desde el Centro de control de DB2 o la línea de mandatos de DB2.

Procedimiento

Para actualizar estadísticas para un apodo individual desde el Centro de control de DB2:

1. Seleccione los apodos necesarios.
 - Aposdos con definiciones de servidor:
 - a. Expanda la carpeta **Objetos de base de datos federada**. Seleccione la carpeta de derivador con la que desee trabajar.
 - b. Expanda la carpeta **Definiciones de servidor**. Seleccione la carpeta de servidor que contenga el apodo con el que desee trabajar.
 - c. Haga una doble pulsación en la carpeta **Apodos**.
 - d. Pulse con el botón derecho del ratón en el nombre del apodo y seleccione **Estadísticas**.
 - e. Seleccione **Actualizar**. Se abrirá el cuaderno Actualización de estadísticas correspondiente a un apodo individual.
 - Aposdos con definición de base de datos:
 - a. Expanda la carpeta **Bases de datos**. Seleccione la carpeta de la **Base de datos** que contenga las definiciones de apodo con las que desee trabajar.
 - b. Haga una doble pulsación en la carpeta **Apodos**.
 - c. Pulse con el botón derecho del ratón en el nombre del apodo que desee actualizar y seleccione **Estadísticas**.
 - d. Seleccione **Actualizar**. Se abrirá el cuaderno Actualización de estadísticas correspondiente a un apodo individual.
2. Especifique los valores para la actualización:
 - Página Actualización de estadísticas:
 - a. Seleccione todas las columnas del apodo para actualizar o seleccione columnas determinadas para actualizar.
 - b. Seleccione todos los índices del apodo para actualizar o seleccione índices determinados para actualizar.
 - c. Seleccione un archivo de registro existente o escriba la vía de acceso totalmente calificada de un nuevo archivo de registro.
 - Página Método. Seleccione uno de los métodos siguientes para la recogida de estadísticas:
 - a. Método basado en catálogos: válido para apodos relacionales.
 - b. Método basado en datos: válido para apodos relacionales y no relacionales.
 - c. Ambos métodos: la opción por omisión.
 - **Opcional:** en la página Planificación, especifique cuándo desea que se ejecute la actualización de estadísticas del apodo.

Obtención de estadísticas de apodo desde la línea de mandatos - ejemplos

Estos ejemplos muestran cómo obtener estadísticas de apodo desde la línea de mandatos utilizando el procedimiento almacenado SYSPROC.NNSTAT.

Ejemplo: Obtener todas las estadísticas

El servidor federado obtiene las estadísticas para los apodos contenidos en el servidor DB2SERV y no crea un archivo de registro.

```
CALL SYSPROC.NNSTAT('DB2SERV',NULL,NULL,NULL,NULL,0,NULL,?)
```

Obtención de todas las estadísticas para un esquema y devolución de un archivo de registro

Ejemplo: Obtención de estadísticas con el método basado en catálogos

El servidor federado obtiene las estadísticas para el apodo STAFF perteneciente al esquema ADMIN. Se recogen estadísticas para las columnas 1 a la 5, y para los índices 1 y 2. Se utiliza el método basado en catálogos para recoger estadísticas. El servidor federado escribe la información del archivo de registro en /home/iiuser/reportlogs/log1.txt.

```
CALL SYSPROC.NNSTAT(  
  NULL, 'ADMIN','STAFF','COL1, COL2, COL3, COL4, COL5','IND1, IND2',1,  
  '/home/iiuser/reportlogs/log1.txt',?)
```

En este ejemplo, el servidor federado recoge estadísticas para todos los apodos contenidos en el servidor DB2Serv y pertenecientes al esquema admin. El servidor federado escribe la información del archivo de registro en /home/iiuser/stats/recent.log.

```
CALL SYSPROC.NNSTAT(  
  'DB2Serv', 'admin', NULL, NULL, NULL, 0, '/home/iiuser/stats/recent.log', ?)
```

Restricciones en las estadísticas de HIGH2KEY y LOW2KEY

Al recoger las estadísticas de HIGH2KEY y LOW2KEY para un apodo de DB2, la información de algunas columnas no se recoge.

Restricciones

Cuando el derivador de DRDA crea un apodo en una tabla o apodo remotos de DB2, sólo recoge las estadísticas de HIGH2KEY y LOW2KEY para columnas numéricas, y sólo si la cardinalidad de la columna es superior a 3.

Procedimiento

Para recoger estadísticas de HIGH2KEY y LOW2KEY, utilice cualquiera de los métodos siguientes:

- Utilice SYSPROC.NNSTAT con el parámetro METHOD establecido en 2. Este valor especifica la recopilación de estadísticas basada en datos. El método basado en datos consulta datos de la tabla remota para calcular los valores de estadísticas locales. Sin embargo, este método puede utilizar recursos significativos del servidor remoto y del servidor federado.
- Emita la sentencia SQL UPDATE para actualizar las columnas HIGH2KEY y LOW2KEY de la vista SYSSTAT.COLUMNS. En este caso, debe determinar manualmente los valores correctos para HIGH2KEY y LOW2KEY.

Creación de un catálogo de herramientas de DB2

Cuando actualiza las estadísticas de un apodo, puede utilizar un catálogo de herramientas de DB2 para planificar las actualizaciones subsiguientes realizadas en estadísticas de apodos. Si carece de un catálogo de herramientas de DB2, se le solicitará que cree un catálogo. Puede crear un catálogo de herramientas de DB2 desde el Centro de control o el indicador de línea de mandatos de DB2, pero para planificar la actualización solamente puede utilizar el Centro de control de DB2.

Antes de empezar

El servidor de administración de DB2 debe estar instalado.

Procedimiento

Para crear un catálogo de herramientas de DB2 desde el Centro de control de DB2:

1. Cuando actualiza estadísticas de apodos se abre la ventana Actualización de estadísticas.
2. Seleccione el sistema donde desee crear una base de datos para el catálogo de herramientas de DB2.

La base de datos debe estar en un sistema catalogado que no tenga almacenamiento de metadatos. Si el sistema elegido no está catalogado, debe catalogarlo antes de crear la base de datos para el catálogo de herramientas de DB2.

Visualización del estado de las actualizaciones hechas en estadísticas de apodo (Centro de control de DB2)

Después de solicitar una actualización de las estadísticas de un apodo, puede ver el estado de la actualización. Puede ver el estado de las actualizaciones hechas en estadísticas de apodo desde el Centro de control de DB2 o la línea de mandatos de DB2.

Procedimiento

Para ver el estado de las actualizaciones hechas en estadísticas de apodo desde el Centro de control de DB2:

1. Seleccione **Actualizar estadísticas**.
2. Seleccione **Ver resultados** y examine la información de estado.

Visualización del estado de las actualizaciones realizadas en estadísticas de apodo (línea de mandatos de DB2)

Después de solicitar una actualización de las estadísticas de un apodo, puede ver el estado de la actualización. Puede ver el estado de las actualizaciones hechas en estadísticas de apodo desde el Centro de control de DB2 o la línea de mandatos de DB2.

Procedimiento

Para ver el estado de las actualizaciones hechas en estadísticas de apodo desde la línea de mandatos de DB2, examine la tabla SYSPROC.FED_STATS.

El ejemplo siguiente muestra una descripción de la tabla SYSPROC.FED_STATS (a efectos de simplificar el ejemplo, la longitud real de las columnas se ha reducido):

```
db2 describe table sysproc.fed_stats
```

Nombre columna	Esquema tipo	Nombre tipo	Long.	Escala	Nulos
SERVER	SYSIBM	VARCHAR	128	0	Sí
SCHEMA	SYSIBM	VARCHAR	128	0	Sí
NICKNAME	SYSIBM	VARCHAR	128	0	Sí
STATS_UPDATE_TIME	SYSIBM	TIMESTAMP	10	0	No
LOG_FILE_PATH	SYSIBM	VARCHAR	1000	0	Sí
SQLCODE	SYSIBM	INTEGER	4	0	Sí
SQLSTATE	SYSIBM	CHARACTER	5	0	Sí
STATUS	SYSIBM	VARCHAR	1000	0	Sí

8 registro(s) seleccionado(s).

```
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 definición de servidor, esquema o apodo especificado no existe.

1 registro(s) seleccionado(s).
.

```

Procedimiento almacenado SYSPROC.NNSTAT

Recuperación de las estadísticas disponibles actualmente para uno o más apodos. Las estadísticas se guardan en el catálogo del sistema contenido en la base de datos federada.

Autorización

SYSPROC.NNSTAT es un procedimiento protegido. El ID de usuario protegido de la instancia federada debe tener autorización para crear el archivo de registro en la ubicación especificada.

Sintaxis

```

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)
)

```

Parámetros

Server Es el servidor donde el servidor federado obtiene las estadísticas de apodo. Este servidor es lo que el usuario registra para definir un origen de datos en la base de datos federada. Si especifica un apodo individual, puede especificar NULL para este parámetro.

Schema

Si especifica NULL para este parámetro, el servidor federado obtiene todos los apodos del servidor especificado. Si el parámetro Servidor es NULL, el servidor federado obtiene las estadísticas del apodo correspondiente al esquema especificado. Si los parámetros Schema y Nickname tienen el valor NULL y se especifica un servidor, el servidor federado obtiene estadísticas en el servidor especificado.

Nickname

El nombre del apodo. Si especifica un apodo, debe también especificar un esquema.

Colnames

Nombres de las columnas, especificados como identificadores de nombres de columna.

Puede especificar este parámetro para un apodo individual solamente. Si especifica nombres de columna, debe también especificar un esquema y un apodo.

Si especifica NULL, se recogen estadísticas para todas las columnas. NULL es el valor por omisión.

Solamente se procesan las columnas especificadas. Si se especifica una cadena de caracteres vacía (""), no se procesa ninguna columna.

Indexnames

Nombres de los índices, especificados como identificadores de nombres de índice.

Puede especificar este parámetro para un apodo individual solamente. Si especifica nombres de índice, debe también especificar un esquema y un apodo. Solamente se procesan los índices especificados.

Si especifica NULL, se recogen estadísticas para todos los índices. NULL es el valor por omisión.

Si se especifica una cadena de caracteres vacía (""), no se procesa ningún índice.

Method

Método utilizado para recoger información estadística a partir del origen de datos.

0 o NULL

Se utiliza primero el método basado en catálogos. Si falla este método, se utiliza el método basado en datos. Esto es el valor por omisión.

1 Recogida de estadísticas basada en catálogos. El método basado en catálogos correlaciona información de catálogos remotos con estadísticas locales correspondientes al apodo. Este método solamente es válido para apodos relacionales.

2 Recogida de estadísticas basada en datos. El método basado en datos consulta datos de la tabla remota para calcular los valores de estadísticas locales. Este método es válido para apodos relacionales y no relacionales.

Este método es la opción por omisión para apodos relacionales si el método basado en catálogos falla para un apodo especificado. La causa habitual por la que no se pueden recoger estadísticas es que los apodos están definidos para vistas remotas. En este caso, no hay estadísticas disponibles en el origen remoto.

Log_File_Path

Nombre y vía de acceso del archivo de registro. El servidor federado crea el archivo de registro en el servidor. Los directorios especificados en la vía de acceso deben existir. En Windows, utilice dos barras inclinadas invertidas para especificar la vía de acceso del archivo de registro. Por ejemplo: c:\\temp\\nnstat.log. Si especifica NULL, el servidor federado no crea un archivo de registro.

Parámetros de salida

out_SQLCode

Error de SQL resultante de la recogida de estadísticas.

out_Trace

El archivo de rastreo.

Ejemplo 1: En este ejemplo, el servidor federado recoge estadísticas para el apodo STAFF perteneciente al esquema ADMIN. Se recogen estadísticas para las columnas 1, 3, 4, 6, 7 y 10, y para los índices 1 al 3. Se utiliza el método basado en datos para recoger estadísticas. El servidor federado escribe la información del archivo de registro en /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',?)
```

Ejemplo 2: En este ejemplo, el servidor federado recoge estadísticas para todos los apodos contenidos en el servidor DB2SERV y pertenecientes al esquema ADMIN. El servidor federado escribe la información del archivo de registro en el archivo c:\reports\log1.txt.

```
CALL SYSPROC.NNSTAT(  
  'DB2SERV','ADMIN',NULL,NULL,NULL,0,'c:\reports\log1.txt',?)
```

Ejemplo 3: En este ejemplo, el servidor federado recoge estadísticas para todos los apodos contenidos en el servidor DB2SERV y no crea un archivo de registro.

```
CALL SYSPROC.NNSTAT(  
  'DB2SERV',NULL,NULL,NULL,NULL,0,NULL,?)
```

Actualización automática de estadísticas de apodos

La recogida automática de estadísticas es una característica que recoge estadísticas actualizadas de tablas y apodos. Esta característica está habilitada por omisión.

Antes de empezar

Para poder ejecutar la recogida automática en un origen de datos, asegúrese de que existe una correlación de usuarios definida para que el propietario de la instancia pueda conectarse al origen de datos.

Acerca de esta tarea

La recogida automática de estadísticas forma parte de la característica de mantenimiento automático de tablas que determina cuándo es necesario actualizar las estadísticas de bases de datos. En el caso de los apodos, la recogida automática de estadísticas utiliza el método basada en catálogos del procedimiento almacenado de estadísticas de apodos (NNSTAT). El método basado en catálogos recupera estadísticas de apodos de la información de catálogos del sitio remoto.

Puede personalizar qué tabla y qué apodos se tienen en cuenta para la recogida automática de estadísticas. Por ejemplo, puede personalizar qué tablas se tienen en cuenta para la característica de recogida automática de estadísticas e incluir o excluir específicamente todos los apodos o los apodos de determinados servidores.

La recogida automática de estadísticas se controla a través de una política de mantenimiento. La misma política se utiliza para la utilidad RUNSTATS automática y para estadísticas sobre apodos. Para crear o personalizar una política, puede utilizar los ejemplos y los archivos de ejemplo que hay disponibles.

Si no desea que las estadísticas sobre apodos se recojan automáticamente, puede inhabilitar la característica de recogida automática de estadísticas o especificar una política para sustituir la política por omisión.

Procedimiento

Para cambiar el comportamiento por omisión de la recogida automática de estadísticas para tabla y apodos:

- Cree una política para las operaciones automáticas de tabla de RUNSTATS. Utilice los procedimientos almacenados del sistema SYSPROC.AUTOMAINT_SET_POLICY y SYSPROC.AUTOMAINT_SET_POLICYFILE.
- Recoja la información sobre políticas de mantenimiento automatizado relacionada con las operaciones automáticas de tabla de RUNSTATS. Utilice los procedimientos almacenados del sistema SYSPROC.AUTOMAINT_GET_POLICY y SYSPROC.AUTOMAINT_GET_POLICYFILE.
- Defina el valor de los parámetros de configuración AUTO_MAINT, AUTO_TBL_MAINT, y AUTO_RUNSTATS en OFF para inhabilitar la recogida automática de estadísticas.

Capítulo 13. Trabajar con datos XML remotos

Federation soporta el tipo de datos XML remotos que ofrece la capacidad de acceder y manipular datos XML en una base de datos DB2 para Linux, UNIX y Windows.

El sistema federado se ajusta a la misma semántica XML que el sistema de base de datos DB2. El almacenamiento de datos XML proporciona la capacidad de almacenar y acceder a los documentos XML almacenados de forma jerárquica como una columna de la tabla relacional. Defina la columna XML con el tipo de datos XML.

Puede crear un apodo relacional en una tabla remota o ver que contiene el tipo de datos XML. También puede utilizar el derivador XML para crear un apodo no relacional en un documento XML.

A continuación, puede utilizar el apodo en lenguaje XQuery y SQL. El lenguaje XQuery es el mecanismo primario para consultar documentos XML. Puede utilizar SQL para realizar operaciones básicas como seleccionar columnas XML o insertar, actualizar o suprimir datos XML. Así mismo, puede combinar SQL y XQuery para crear consultas para datos relacionales existentes y datos XML utilizando funciones y predicados SQL/XML y funciones XQuery.

Creación de un apodo para datos XML remotos - ejemplos

Para trabajar con datos XML remotos, es necesario crear un apodo para una tabla remota que contenga datos XML o para un documento XML.

Ejemplo: cree un apodo relacional que corresponda a una tabla de DB2 remota llamada XMLTABLE1. La tabla XMLTABLE1 contiene la columna XMLCOL definida con el tipo de datos XML:

```
CREATE NICKNAME NNXML1 FOR SERVER1.SCHEMA1.XMLTABLE1;
```

Ejemplo: cree un apodo no relacional para un documento XML utilizando el derivador XML:

```
CREATE NICKNAME NNXML2
  (file_path VARCHAR(64) OPTIONS(DOCUMENT 'FILE'),
   doc XML)
FOR SERVER XML_SERVER;
```

Manipulación de datos XML - ejemplos

Después de crear un apodo, podrá consultar los datos XML en la tabla remota o en el documento XML del sistema federado. El sistema federado soporta todas las operaciones de manipulación de datos para los datos XML que soporta el sistema de base de datos DB2.

Los ejemplos siguientes muestran los métodos que puede utilizar para trabajar con datos XML utilizando el apodo NNXML1.

Manipulación de datos XML con SQL

Con SQL puede realizar operaciones básicas de selección, inserción, actualización y supresión.

Ejemplo: utilice las sentencias SELECT e INSERT para acceder e insertar datos XML desde el apodo NNXML1.

En la siguiente sentencia SELECT, el documento XML se selecciona en el servidor federado:

```
SELECT xmlcol FROM NNXML1;
```

En la siguiente sentencia INSERT se inserta una serie en la columna XML de apodo:

```
INSERT INTO NNXML1 (xmlcol) VALUES ('<a><b>My data</b></a>');
```

Manipulación de datos XML con funciones SQL/XML

Con funciones SQL/XML, podrá realizar múltiples operaciones como consultar, validar y publicar datos XML.

Ejemplo: Utilice la función XMLVALIDATE para validar datos XML utilizando el esquema XML obtenido de la especificación de esquemas en el documento de instancia XML:

```
SELECT XMLVALIDATE(xmlcol) FROM NNXML1;
```

Manipulación de datos XML con XQuery

Puede utilizar funciones XQuery para manipular datos XML, como por ejemplo la función xmlcolumn.

Ejemplo: Utilice XQuery para recuperar el elemento b, hijo de la raíz a, de la columna XML llamada XMLCOL:

```
xquery db2-fn:xmlcolumn('NNXML1.XMLCOL')/a/b;
```

Validación de documentos XML contra esquemas XML

Puede verificar si un documento XML es válido. Para validar un documento XML, es necesario registrar el esquema XML e invocar explícitamente la función XMLVALIDATE. La validación es opcional, pero recomendada.

Registro de esquemas XML

El registro de esquemas es necesario para la validación XML. También se debe registrar el esquema antes de realizar descomposiciones de esquema anotado.

Acerca de esta tarea

Registre un esquema XML en el repositorio de esquemas XML (XSR). El proceso de registro creará un objeto XSR.

El registro de esquema XML se produce en el servidor federado.

Procedimiento

Para registrar objetos XSR:

1. Registre el documento de esquema XML en XSR utilizando procedimientos almacenados o mandatos a través del procesador de la línea de mandatos.
2. Especifique documentos de esquema XML adicionales para incluir el objeto XSR si el esquema XML consta de más de un documento de esquema.
3. Complete el proceso de registro con XSR.

Validación de documentos XML

Para validar un documento XML concreto, se debe invocar explícitamente la función XMLVALIDATE.

Acerca de esta tarea

Normalmente los datos XML se validan durante una operación de inserción o actualización utilizando la función XMLVALIDATE. La función XMLVALIDATE valida un valor XML contra un esquema XML o contra el esquema XML obtenido de la especificación de esquemas en el documento de instancia.

Cuando se inserta o se actualiza una columna XML en un apodo y se utiliza la función XMLVALIDATE, se produce la validación en el servidor federado. A continuación, el servidor federado serializa los dtos y los envía al origen de datos. Durante la serialización, no se conserva ninguna anotación de tipo de datos añadida por XMLVALIDATE.

Procedimiento

Para validar un documento XML:

Invoque la función XMLVALIDATE como parte de una sentencia SQL.

- Si se llama a XMLVALIDATE sin una especificación de esquema, el esquema se determina basándose en el atributo *schemaLocation* en el documento de instancia. Si no existe la información de esquema, se emitirá un mensaje de error.
- Si se llama a XMLVALIDATE con un esquema registrado o con un identificador universal de recursos (URI), la validación se realizará contra el esquema específico. Si se especifica un esquema externo, éste sustituirá al esquema interno.

Ejemplo: Valide un documento XML utilizando el esquema XML llamado MYSCHEMA.MYDOCUMENTS:

```
INSERT INTO NXML1(XMLCOL)
VALUES (
XMLVALIDATE(
? ACCORDING TO XMLSCHEMA ID MYSCHEMA.MYDOCUMENTS
)
)
```

Si el esquema XML que está asociado con el identificador SQL MYSCHEMA.MYDOCUMENTS existe en el repositorio XML, se validará el valor XML de entrada.

Descomposición de documentos XML con esquemas XML anotados en apodos

Con la descomposición de esquema XML anotado, se pueden descomponer documentos basados en anotaciones especificadas en un esquema XML.

Antes de empezar

Los documentos de esquema anotado se deben almacenar y registrar con el repositorio de esquema XML (XSR).

Acerca de esta tarea

En los casos en que es necesario acceder a datos XML como datos relacionales, en lugar de hacerlo de forma jerárquica, se pueden descomponer o reducir. Un documento XML se puede descomponer en un apodo que haga referencia a una tabla remota.

Procedimiento

Para descomponer documentos XML con esquemas XML anotados en apodo:

1. Anote los documentos de esquema con una descomposición de esquema XML
2. Cree el apodo que desee utilizar para la descomposición. El nombre del apodo debe coincidir con los valores del esquema anotado.
3. Registre el esquema utilizando el mandato XMLSCHEMA.
4. Otorgue privilegios USAGE en el objeto XSR para permitir que usuarios específicos utilicen un esquema XML concreto.
5. Habilite el esquema para la descomposición utilizando la sentencia ALTER XSROBJECT.
6. Utilice el mandato DECOMPOSE XML DOCUMENT para descomponer el documento de instancia XML.

Proceso federado de datos XML remotos

Los datos XML se envían y reciben entre el sistema federado y el cliente de origen de datos remotos como una serie XML serializada en formato de caracteres o en formato binario.

Los procesos que afectan al modo en que el servidor federado envía y recibe datos XML son la serialización, el análisis y los convenios de página de códigos.

Análisis federado de datos XML remotos

El sistema federado utiliza el analizador XML de DB2 para procesar datos XML remotos.

El analizador necesita datos XML formados correctamente que sigan las normas de codificación de datos. El analizador no realiza la validación de esquemas.

- Durante la recuperación de datos, el analizador emite un mensaje de error si los datos XML no se han formado correctamente.
- Durante la inserción de datos, en función de dónde se realice el análisis, el servidor federado o el origen de datos de destino emite un mensaje de error si los datos XML no se han formado correctamente.

Manejo de espacios en blanco para Federation

Durante un análisis XML, puede conservar o eliminar los espacios en blanco limitadores en los documentos XML. Los caracteres de espacio en blanco que se producen entre elementos son *espacios en blanco limitadores*.

El servidor federado refuerza los mismos convenios de espacio en blanco que el sistema de base de datos DB2.

Para variables de host de aplicación y marcadores de parámetro, el registro especial CURRENT IMPLICIT XMLPARSE OPTION determina si el espacio en blanco se eliminará durante la operación de vinculación federada. Si el origen de datos sigue normas distintas para el manejo de espacios en blanco, el sistema federado intentará compensar la diferencia semántica.

Problemas de página de códigos para datos XML remotos

Ciertos problemas de página de códigos pueden afectar a las sentencias federadas.

El servidor federado cumple las normas de codificación del analizador XML de DB2.

Para sentencias federadas, pueden producirse los problemas siguientes al manipular datos XML remotos serializados:

- Para datos XML remotos serializados en formato binario:
 - Cuando se envían los datos desde el cliente remoto al derivador federado, no se produce ninguna pérdida de datos.
 - Si existe una codificación interna que no es un esquema de codificación de IBM válido, el derivador sustituye el atributo de codificación con un esquema de codificación de IBM válido o elimina el atributo de codificación de la declaración XML y el analizador XML de DB2 descodifica el valor.
- Para datos XML remotos serializados en formato de caracteres:
 - La página de códigos de los datos XML está en la página de códigos de la base de datos federada. Cuando el cliente del origen de datos envía datos al derivador federado, es posible que se produzca una pérdida de datos. Si la conversión resulta en una sustitución de carácter, debe generarse un aviso en función del comportamiento del origen de datos y de la implementación del derivador.
 - Si existe una codificación interna, el derivador la eliminará porque la serie serializada de la página de códigos está en la página de códigos de la base de datos federada.

Para derivadores no relacionales, el analizador XML de DB2 descodificará los datos XML utilizando la codificación interna del documento.

Restricciones sobre tipos de datos XML remotos

Los sistemas federados imponen restricciones sobre el tipo de datos XML remotos.

Sólo se puede utilizar el tipo de datos XML con el origen de datos y el derivador XML de DB2 para Linux, UNIX y Windows.

No se pueden realizar las operaciones SQL siguientes:

- Al alterar el tipo de datos XML para un apodo, se produce un error SQL0270N

- Al crear un índice federado SPECIFICATION ONLY con la cláusula de especificación de índice xml, se produce un error SQL0104N
- Al definir un procedimiento almacenado federado con el tipo de datos XML, se produce un error SQL1254N
- Al crear una correlación de tipo de datos entre el tipo de datos XML y otro tipo de datos, se produce un error SQL0604N

Las restricciones siguientes se aplican sólo a las columnas XML para el derivador XML:

- No se puede especificar ninguna opción de columna para una columna XML.
- Sólo el apodo raíz puede contener una única columna XML que haga referencia a todo el contenido de un documento XML.
- Cada apodo raíz debe corresponder a un único documento XML.
- Las opciones de los apodos hijo, las opciones XPath de apodo y las opciones de espacio de nombre de apodo no son importantes.

Capítulo 14. Tolerancia a errores en expresiones de tabla anidadas

La tolerancia a errores es un mecanismo que permite proseguir la ejecución de consultas aunque se produzcan determinados errores de SQL en expresiones de tabla anidadas. Gracias a la tolerancia a errores, en lugar de recibir un error para una parte de una consulta e interrumpir la consulta completa, el usuario puede obtener la máxima cantidad de resultados de la consulta a partir de los datos disponibles.

Cuando el servidor federado encuentra un error permisible, el servidor permite el error y prosigue el proceso del resto de la consulta en lugar de devolver un error para la consulta completa. El resultado devuelto por el servidor federado puede ser un resultado parcial o un resultado vacío.

Cuando el servidor federado tolera errores, devuelve resultados de la consulta aunque los orígenes de datos a los que accede la consulta no estén disponibles. Este mecanismo es útil cuando es necesario obtener el máximo de información disponible, aunque los resultados de la consulta sean incompletos. Por ejemplo, considere el caso de un doctor que necesita datos sobre un tipo determinado de condición médica. Se ejecuta una consulta que recoge información procedente de orígenes de datos situados en diversos hospitales. Si una o más bases de datos hospitalarias no están disponibles, los resultados procedentes solamente de las bases de datos disponibles seguirán siendo muy valiosos para el doctor.

Las consultas que contienen ramas UNION ALL pueden sacar provecho de la tolerancia a errores. Sin este mecanismo, si el proceso de una rama de la consulta encuentra un error, el servidor federado detiene el proceso de la consulta. Gracias a la tolerancia a errores, cuando especifica ese mecanismo para esa rama de la consulta, el servidor federado tolera el error y prosigue el proceso de las demás ramas disponibles. La operación UNION ALL devuelve resultados a partir de cualquier origen de datos disponible.

Ejemplo: la consulta siguiente selecciona datos de tres apodos o tres servidores diferentes:

```
SELECT c1 from nickname1_on_server1
UNION ALL
SELECT c1 from nickname2_on_server2
UNION ALL
SELECT c1 from nickname3_on_server3
```

Cuando nickname2_on_server2 no está disponible o cuando el servidor remoto server2 no está disponible durante el proceso de la consulta, la tolerancia a errores en nickname2 y server2 permite obtener resultados a partir de nickname1_on_server1 y nickname3_on_server3. El resultado obtenido a partir de dos de las tres ramas de la consulta es equivalente a ejecutar la consulta siguiente:

```
SELECT c1 from nickname1_on_server1
UNION ALL
SELECT c1 from nickname3_on_server3
```

El usuario puede especificar los errores de SQL que desea permitir en una expresión de tabla anidada durante el proceso de la consulta. Los tipos de errores

que el servidor federado tolera son los errores relacionados con las conexiones remotas, la autorización y la autenticación.

Especificación de expresiones de tabla anidadas para la tolerancia a errores

Utilice una expresión de tabla anidada con la cláusula RETURN DATA UNTIL para especificar los errores que se deben tolerar.

Acerca de esta tarea

Cuando utiliza la cláusula RETURN DATA UNTIL, debe especificar los códigos de error que desea tolerar. La tabla siguiente lista los errores que están permitidos en la cláusula *valor-condición-específica*. Debe especificar un SQLSTATE, o un SQLSTATE y SQLCODE, que coincida con un código de error permisible. Los SQLCODE listados en la tabla son obligatorios.

Tabla 19. Errores permitidos en la cláusula *valor-condición-específica*

SQLSTATE	Código de error	SQLCODE
08001	SQL30080N	-30080
08001	SQL30081N	-30081
08001	SQL30082N	-30082
08001	SQL1336N	-1336
08004	Cualquiera	Cualquiera
28000	Cualquiera	Cualquiera
42501	Cualquiera	Cualquiera
42512	Cualquiera	Cualquiera
42704	SQL0204N	-204
42720	Cualquiera	Cualquiera

Procedimiento

Para especificar expresiones de tabla anidadas para la tolerancia a errores, cree una sentencia de SQL que contenga la cláusula RETURN DATA UNTIL.

RETURN DATA UNTIL *valor-condición-específica*

RETURN DATA UNTIL

Los resultados de la selección completa incluyen todas las filas que devolvió la operación hasta que encontró la condición especificada.

valor-condición-específica

Especifica la condición y los valores utilizados para la tolerancia a errores.

FEDERATED

Palabra clave obligatoria. La condición específica que especifique debe incluir solamente los errores producidos en un origen de datos federado.

SQLSTATE VALUE *constante-tipo-serie*

Puede especificar una condición específica como valor de SQLSTATE. La longitud de la constante de tipo serie debe ser 5 cuando se especifica VALUE. El valor de SQLSTATE se puede restringir para un determinado valor de SQLCODE. En una

cláusula *valor-condición-específica* puede especificar varios valores de SQLCODE que compartan el mismo SQLSTATE.

Expresiones de tabla anidadas para la tolerancia a errores - ejemplo

Los ejemplos siguientes muestran cómo utilizar la cláusula RETURN DATA UNTIL para obtener los resultados de una consulta cuando no están disponibles uno o más orígenes de datos.

Ejemplo: la sentencia de SQL siguiente selecciona datos de tres servidores diferentes: SQLServer, Oracle y 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;
```

Caso 1: no está disponible un servidor.

En este caso, el servidor Oracle no está disponible. El servidor SQLServer y el servidor Sybase están disponibles. En esta situación, la consulta contenida en la segunda rama de la operación UNION ALL devuelve un resultado vacío y el error 30080, para el cual se ha especificado que debe ser tolerado. La consulta devuelve resultados procedentes de n1_from_SQLServer y n3_from_Sybase. Se emite el aviso sqlwarn5='E'.

El resultado es equivalente a ejecutar la consulta siguiente:

```
SELECT c1 FROM n1_from_SQLServer
UNION ALL
SELECT c1 FROM n3_from_Sybase;
```

Caso 2: no está disponible ningún servidor.

En este caso, todos los servidores (SQLServer, Oracle y Sybase) no están disponibles. En esta situación, la operación UNION ALL devuelve un resultado vacío. Se emite el aviso sqlwarn5='E'.

Caso 3: todos los servidores están disponibles.

Si todos los servidores están disponibles, el resultado de la consulta es equivalente a ejecutar la misma consulta sin especificar la cláusula RETURN DATA UNTIL.

Soporte de orígenes de datos para expresiones de tabla anidadas para la tolerancia a errores

La tolerancia a errores se puede utilizar con varios orígenes de datos relacionales y con apodos no relacionales.

La tolerancia a errores en expresiones de tabla anidadas es compatible con los siguientes orígenes de datos relacionales:

- Familia de productos DB2(DRDA)
- Informix
- JDBC
- Microsoft SQL Server
- ODBC
- Oracle (NET8)
- Sybase (CTLIB)
- Teradata

Puede utilizar apodos no relacionales dentro de expresiones de tabla anidadas para la tolerancia a errores. El servidor federado puede tolerar los errores permisibles de conexión, autenticación o autorización cuando los derivadores no relacionales devuelven un código de error permisible.

Restricciones respecto a expresiones de tabla anidadas para la tolerancia a errores

Son aplicables algunas restricciones cuando define expresiones de tabla anidadas para la tolerancia a errores.

Una consulta o vista es de sólo lectura cuando define la consulta o vista con una expresión donde está contenida una cláusula RETURN DATA UNTIL. Los cursores que se declaran en expresiones que contienen la cláusula RETURN DATA UNTIL son cursores de sólo lectura. Se devuelven errores para cada una de esas situaciones.

Capítulo 15. Supervisión de apodos y servidores federados

Puede utilizar el supervisor de estado y elementos del supervisor de sistemas para supervisar un sistema federado.

Indicadores de salud para servidores y apodos federados

Puede utilizar indicadores de salud en el Centro de salud de DB2 para supervisar el estado de los servidores y los apodos federados.

El indicador de salud para apodos es `db.fed_nicknames_op_status`. El indicador de salud para definiciones de servidor es `db.fed_servers_op_status`. Los indicadores de salud federados están instalados cuando se instala el supervisor de estado.

Por defecto, el Centro de salud no activa los indicadores de salud federados. Debe activar los indicadores.

Cuando el estado de un apodo o de un servidor no es normal, los indicadores de salud emiten una aviso. Puede ver los resultados de la supervisión utilizando el Centro de salud o la línea de mandatos.

Los servidores federados que utilizan los sistemas operativos AIX, HP-UX, Linux, Microsoft Windows y Solaris soportan los indicadores de salud.

Tabla 20 describe los indicadores de salud para los servidores y apodos federados.

Tabla 20. Indicadores de salud para servidores y apodos

Indicador de salud	Descripción
<code>db.fed_nicknames_op_status</code>	<p>Indica el estado de todos los apodos relacionales definidos en una base de datos o en un servidor federado.</p> <p>Emite un aviso si un apodo no es válido. Proporciona detalles sobre los apodos no válidos y recomienda acciones que se pueden emprender para arreglarlos.</p>
<code>db.fed_servers_op_status</code>	<p>Indica el estado de todos los servidores federados definidos en una base de datos o en un servidor federado.</p> <p>Emite un aviso si un servidor no está disponible. Proporciona detalles sobre los servidores que no están disponibles y recomienda acciones que se pueden emprender para hacer que estén disponibles.</p>

Los indicadores de salud pueden evaluar los orígenes de datos siguientes:

- Familia de productos DB2(DRDA)
- Excel
- Informix
- JDBC
- Microsoft SQL Server
- ODBC

- Oracle (NET8)
- Sybase (CTLIB)
- Archivos con estructura de tabla
- Teradata
- XML (sólo apodos raíz)

Activación de los indicadores de salud federados

Para supervisar el estado de los servidores y de los apodos, debe activar los indicadores de salud federados. El indicador de salud para apodos es `db.fed_nicknames_op_status`. El indicador de salud para definiciones de servidor es `db.fed_servers_op_status`.

Procedimiento

Para activar los indicadores de salud federados, puede abrir el Centro de salud de DB2 y configurar los indicadores de salud o utilizar el procesador de línea de mandatos.

Supervisión de estado de los servidores y los apodos federados

Si supervisa el estado de servidores y apodos le puede ayudar a determinar y resolver problemas del sistema federado. Puede supervisar el estado de los servidores y apodos federados utilizando indicadores de salud en el Centro de salud.

Antes de empezar

- Asegúrese de que se han definido privilegios SELECT en los apodos del servidor federado.
- Establezca el parámetro de configuración del gestor de base de datos FEDERATED en YES.
- Si el origen de datos requiere autenticación, el origen de datos debe tener correlaciones de usuario del ID del supervisor de estado. El supervisor de estado utiliza esta correlación para conectarse al origen de datos.

Restricciones

“Indicadores de salud para servidores y apodos federados” en la página 197 hace una lista de los orígenes de datos que pueden evaluar los indicadores de salud.

Acerca de esta tarea

Puede ver los resultados de la supervisión utilizando el Centro de salud o la línea de mandatos. Utilice el Centro de control de DB2 o el procesador de línea de mandatos de DB2 para resolver los problemas que han identificado los indicadores de salud.

Procedimiento

Para efectuar esta tarea desde la línea de mandatos, emita el mandato GET HEALTH SNAPSHOT.

Para realizar esta tarea desde el Centro de control de DB2:

1. Abra el Centro de salud.

2. Abra el Asesor de recomendaciones para ver recomendaciones sobre cómo resolver los apodos no válidos o los servidores no disponibles.
3. Para efectuar esta tarea desde la línea de mandatos, emita el mandato GET HEALTH SNAPSHOT.

Supervisión de estado de los servidores y los apodos federados - ejemplo

Este tema proporciona un ejemplo de una instantánea de estado de una base de datos.

Los nombres de los indicadores de salud federados son db.fed_nicknames_op_status y db.fed_servers_op_status. Debe habilitar estos indicadores utilizando el Centro de salud o bien los siguientes mandatos en CLP:

```
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
```

El siguiente mandato recupera una instantánea de estado de base de datos incluyendo los indicadores de salud federados, si es que se han habilitado:

```
db2 get health snapshot for database on <database_name>
```

En este ejemplo, el nombre de la base de datos es fedhi. La salida de este mandato indica que ambos indicadores de salud están en estado normal. Normal significa que los apodos y los servidores son válidos.

Database Health Snapshot

```
Snapshot timestamp                = 02/10/2006 12:10:55.063004
Database name                     = FEDHI
Database path                     = C:\DB2\NODE0000\SQL00006\
Input database alias              = FEDHI
Operating system running at database server= NT
Location of the database          = Local
Database highest severity alert state = Attention
```

Health Indicators:

```
Indicator Name                    = db.fed_servers_op_status
Value                             = 0
Evaluation timestamp              = 02/10/2006 12:09:10.961000
Alert state                       = Normal
```

```
Indicator Name                    = db.fed_nicknames_op_status
Value                             = 0
Evaluation timestamp              = 02/10/2006 12:09:10.961000
Alert state                       = Normal
```

```
Indicator Name                    = db.db_op_status
Value                             = 0
Evaluation timestamp              = 02/10/2006 12:08:10.774000
Alert state                       = Normal
```

```
Indicator Name                    = db.sort_shrmem_util
Value                             = 0
Unit                              = %
Evaluation timestamp              = 02/10/2006 12:08:10.774000
Alert state                       = Normal
```

```
Indicator Name                    = db.spilled_sorts
```

Value	= 0
Unit	= %
Evaluation timestamp	= 02/10/2006 12:09:10.961000
Alert state	= Normal

Supervisión de instantánea de sistemas federados - Visión general

Puede utilizar el supervisor de instantáneas para obtener información sobre orígenes de datos federados y de las aplicaciones conectadas en un momento concreto.

Las instantáneas son útiles para determinar el estado de un sistema federado. En intervalos regulares, las instantáneas también son útiles para observar tendencias y prever posibles problemas.

Las salida del supervisor de instantáneas está disponible en los formatos siguientes:

- En formato textual, a través de la interfaz del procesador de línea de mandatos del supervisor de instantáneas.
- Como salida de funciones de tablas. Esta salida es útil para escribir consultas que limitan la salida.

Las instantáneas que son especialmente útiles para las cargas de trabajo federadas son:

Instantánea de sentencia de SQL dinámico

Proporciona una instantánea para todas las sentencias de SQL dinámico que actualmente están en la caché de sentencia y que incluyen sentencias federadas y no federadas.

Instantánea de aplicación

Proporciona información sobre una aplicación concreta, incluyendo el texto de la sentencia de SQL que está actualmente en funcionamiento.

Instantánea de bases de datos remota

Proporciona información sobre una base de datos específica del sistema federado.

Todas las instantáneas de bases de datos remotas

Proporciona información sobre cada base de datos del sistema federado que está activa.

Instantáneas de aplicaciones remotas

Proporciona información a nivel de aplicación para cada aplicación del sistema federado que está activa.

Supervisión de consultas federadas

Si supervisa consultas, puede determinar cómo está funcionando el sistema federado. Para ayudarle a entender cómo el sistema federado está procesando una consulta, puede obtener una instantánea de una consulta remota.

Acerca de esta tarea

El supervisor de instantáneas supervisa dos aspectos de cada consulta que procesa el servidor federado:

- La aplicación envía la consulta federada completa, que hace referencia a apodos, tablas locales o ambas cosas.

- Para las consultas que hacen referencia a apodos, uno o más *fragmentos remotos*. Los fragmentos remotos son sentencias que se generan y envían automáticamente a orígenes de datos remotos en sus dialectos nativos en nombre de la consulta federada.

Para la supervisión de consultas federadas es necesario que tenga en cuenta el trabajo que ha realizado a nivel local en el servidor federado y el trabajo realizado en los servidores remotos como respuesta a fragmentos de consulta remotos. La instantánea de sentencia de SQL dinámico y la función de tabla `SNAPSHOT_DYN_SQL` contienen información sobre consultas federadas individuales cuando se envían al servidor federado, y sobre los fragmentos de consulta remotos que el servidor federado envía a otros orígenes de datos.

Antes de empezar

Debe establecer el supervisor `STATEMENT` en `ON` para que la base de datos federada recoja información de instantánea para las consultas remotas.

Procedimiento

Para supervisar consultas en el servidor federado, mientras está conectado a la base de datos federada, utilice uno de los siguientes métodos:

- Salida textual:

```
GET SNAPSHOT FOR DYNAMIC SQL on dbname
```

donde *dbname* es el nombre de la base de datos del sistema federado.

- Función de tabla:

```
CREATE TABLE table_snap AS (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL ('dbname', -1))
  as snaptab) definition only;
INSERT INTO snap (SELECT * FROM TABLE(SNAPSHOT_DYN_SQL ('dbname', -1))
  as snaptab);
```

Puede escribir una consulta sobre la tabla que contiene una fila por consulta (federada o no federada) y una fila por fragmento de consulta en la caché de sentencias del servidor.

El nombre de los fragmentos de consulta remotos es el servidor al que se envió el texto de la consulta remota, entre corchetes, en el campo de la función de tabla `stmt_text`. Por ejemplo, puede utilizar la consulta siguiente para buscar fragmentos remotos de larga ejecución:

```
SELECT total_exec_time, rows_read, total_usr_cpu_time, num_executions,
  substr(stmt_text,1,30)
FROM TABLE(SNAPSHOT_DYN_SQL ('dbname', -1))AS snaptab
-- remote fragments only
WHERE stmt_text LIKE '[%]%'
ORDER BY total_exec_time;
```

Si se compara el tiempo de ejecución de una sentencia federada completa con los tiempos de ejecución de los fragmentos remotos enviados a otros orígenes de datos en nombre de la sentencia, podrá ver donde se gasta más tiempo de proceso.

para determinar qué fragmentos de consulta se envían a los orígenes remotos en nombre de una consulta federada, consulte el plan de ejecución `EXPLAIN` de la consulta.

Supervisión de instantánea de consultas federadas - Visión general

Este tema proporciona un ejemplo para una instantánea de SQL dinámico basada en texto de una consulta federada que implica un origen de datos remoto de Oracle.

La siguiente sentencia recupera una instantánea de todas las sentencias que actualmente están en la caché de sentencias, incluyendo sentencias federadas y fragmentos remotos enviados a otros orígenes de datos:

```
GET SNAPSHOT FOR DYNAMIC SQL ON <database_name>
```

El nombre de la base de datos es el nombre de la base de datos federada local.

La salida del siguiente ejemplo es el resultado de la sentencia:

```
GET SNAPSHOT FOR DYNAMIC SQL ON FEDDB
```

El ejemplo muestra una sentencia federada y un fragmento remoto que emite la sentencia federada. Puede identificar los fragmentos remotos buscando el nombre del servidor necesario para el texto de la sentencia remoto que está entre corchetes. En este ejemplo, el servidor remoto de Oracle se llama ORA9. La primera entrada muestra la sentencia de SQL federada que hace referencia a apodos, incluyendo todo el tiempo que ha transcurrido. La segunda entrada muestra la sentencia remota enviada al origen [ORA9] que hace referencia a nombres de tablas de Oracle remotas.

```
Dynamic SQL Snapshot Result

Database name                = FEDDB

Number of executions         = 1
Number of compilations       = 1
Worst preparation time (ms)  = 475
Best preparation time (ms)   = 475
Internal rows deleted        = 0
Internal rows inserted       = 0
Rows read                    = 5
Internal rows updated        = 0
Rows written                 = 0
Statement sorts              = 0
Statement sort overflows     = 0
Total sort time              = 0
Buffer pool data logical reads = Not Collected
Buffer pool data physical reads = Not Collected
Buffer pool temporary data logical reads = Not Collected
Buffer pool temporary data physical reads = Not Collected
Buffer pool index logical reads = Not Collected
Buffer pool index physical reads = Not Collected
Buffer pool temporary index logical reads = Not Collected
Buffer pool temporary index physical reads = Not Collected
Buffer pool xda logical reads = Not Collected
Buffer pool xda physical reads = Not Collected
Buffer pool temporary xda logical reads = Not Collected
Buffer pool temporary xda physical reads = Not Collected
Total execution time (sec.ms) = 1.816884
Total user cpu time (sec.ms) = 0.000000
Total system cpu time (sec.ms) = 0.020000
Statement text                = select count(*) from orat.supplier,
                             orat.nation where s_nationkey =
                             n_nationkey and n_name <> 'FRANCE'

Number of executions         = 1
```

```

Number of compilations           = 1
Worst preparation time (ms)      = 0
Best preparation time (ms)       = 0
Internal rows deleted            = 0
Internal rows inserted          = 0
Rows read                        = 1
Internal rows updated            = 0
Rows written                     = 0
Statement sorts                  = 0
Statement sort overflows        = 0
Total sort time                  = 0
Buffer pool data logical reads   = Not Collected
Buffer pool data physical reads  = Not Collected
Buffer pool temporary data logical reads = Not Collected
Buffer pool temporary data physical reads = Not Collected
Buffer pool index logical reads  = Not Collected
Buffer pool index physical reads = Not Collected
Buffer pool temporary index logical reads = Not Collected
Buffer pool temporary index physical reads = Not Collected
Buffer pool xda logical reads    = Not Collected
Buffer pool xda physical reads   = Not Collected
Buffer pool temporary xda logical reads = Not Collected
Buffer pool temporary xda physical reads = Not Collected
Total execution time (sec.ms)    = 1.337672
Total user cpu time (sec.ms)     = 0.000000
Total system cpu time (sec.ms)   = 0.000000
Statement text                    = [ORA9] SELECT COUNT(*) FROM "TPCH"."NATION"
                                A0, "TPCH"."SUPPLIER" A1 WHERE
                                (A0."N_NAME" <> 'FRANCE                ') AND
                                (A1."S_NATIONKEY" = A0."N_NATIONKEY")

```

La instantánea no ha recogido ninguna información de la agrupación de la memoria intermedia, porque la información de la agrupación de la memoria intermedia no se puede aplicar a consultas remotas.

Elementos del supervisor de sistemas de base de datos federada

Este tema describe los elementos del supervisor que proporcionan información sobre sistemas federados.

Un sistema federado accede a varios orígenes de datos que pueden encontrarse en distintas plataformas, tanto de IBM como de otros proveedores, relacionales o no relacionales. Integra el acceso a datos distribuidos y presenta una sola imagen de base de datos de un entorno heterogéneo a sus usuarios.

Los siguientes elementos muestran información sobre el acceso total a un origen de datos mediante aplicaciones que se ejecutan en un sistema federado e información sobre cómo acceder a un origen de datos mediante una aplicación que se ejecute en una instancia de servidor federada. Incluyen:

- `datasource_name` - Elemento de supervisor Nombre de origen de datos
- `disconnects` - Elemento de supervisor Desconecta
- `insert_sql_stmts` - Elemento de supervisor Inserta
- `update_sql_stmts` - Elemento de supervisor Actualiza
- `delete_sql_stmts` - Elemento de supervisor Suprime
- `dynamic_sql_stmts` - Elemento de supervisor Sentencias SQL dinámicas realizadas
- `create_nickname` - Elemento de supervisor Crear apodos
- `passthru` - Elemento de supervisor Paso a través
- `stored_procs` - Elemento de supervisor Procedimientos almacenados

- remote_locks - Elemento de supervisor Bloqueos remotos
- sp_rows_selected - Elemento de supervisor Filas devueltas por procedimientos almacenados
- select_time - Elemento de supervisor Tiempo de respuesta de consulta
- insert_time - Elemento de supervisor Insertar tiempo de respuesta
- update_time - Elemento de supervisor Actualizar tiempo de respuesta
- delete_time - Elemento de supervisor Suprimir tiempo de respuesta
- create_nickname_time - Elemento de supervisor Tiempo de respuesta de apodo
- passthru_time - Elemento de supervisor Tiempo de paso a través
- stored_proc_time - Elemento de supervisor Tiempo de procedimientos almacenados
- remote_lock_time - Elemento de supervisor Tiempo de bloqueo remoto

El siguiente ejemplo muestra una instantánea dynamic_sql_statement:

```
Statement text = [ORACLE817]SELECT A0.C1,A0.C2 FROM ORA_T A0 WHERE A0.C3 = :H0
```

Para todas las sentencias remotas, el texto de la sentencia (Statement text) empieza con el nombre del origen de datos remotos, entre corchetes, seguido por el texto real que se envía al origen de datos remoto.

Capítulo 16. Cómo interactúan las aplicaciones cliente con los orígenes de datos

Para las aplicaciones cliente, los orígenes de datos de un sistema federado aparecen como una única base de datos colectiva. Para obtener datos de los orígenes de datos, las aplicaciones envían consultas en SQL de DB2 a la base de datos federada. La base de datos federada distribuye entonces las consultas a los orígenes de datos apropiados y devuelve estos datos a las aplicaciones o realiza la acción solicitada.

La base de datos federada puede unir datos desde tablas locales y orígenes de datos remotos en la misma sentencia SQL. Por ejemplo, puede unir datos que estén ubicados en una tabla DB2 local, una tabla Informix y una vista Sybase en una única sentencia SQL. Procesando sentencias SQL como si los orígenes de datos fueran vistas o tablas relacionales ordinarias en la base de datos federada, el sistema federado puede unir datos relacionales y datos no relacionales.

En un sistema federado, podrá acceder a orígenes de datos por medio de apodos. Un *apodo* es un objeto de base de datos federada que una aplicación utiliza para hacer referencia a un objeto de origen de datos, como por ejemplo una tabla o vista. Para grabar en un origen de datos, por ejemplo, para actualizar una tabla de origen de datos, una aplicación puede utilizar SQL de DB2 (con apodos). Opcionalmente, las aplicaciones pueden utilizar el dialecto de SQL del origen de datos (sin apodos) en una sesión especial llamada *paso a través* para acceder directamente a los orígenes de datos.

Las aplicaciones que utilizan apodos y SQL de DB2 pueden acceder a cualquier tipo de datos que reconozca la base de datos federada.

El catálogo de la base de datos federada contiene información acerca de los objetos de la base de datos federada e información acerca de los objetos de los orígenes de datos. Puesto que el catálogo contiene información sobre la base de datos federada completa, se le denomina *catálogo global*.

Capítulo 17. Referencia a objetos de origen de datos por medio de apodos en sentencias SQL

Con un sistema federado, utilizará los apodos definidos para los objetos de origen de datos para representar los objetos en las sentencias SQL. El sistema federado no reconoce nombres de objeto, esquema y origen de datos completamente calificados en sentencias SQL.

Los objetos de origen de datos deben tener apodos registrados en la base de datos federada antes de que puedan incluirse en las consultas. En general, podrá especificar apodos en una sentencia SQL en la que pueda especificar tablas locales en una sentencia SQL.

Ejemplo: Utilización de apodos en las sentencias SELECT, INSERT, UPDATE y DELETE

Defina el apodo NFXDEPT para que represente una tabla en una tabla Informix denominada PERSON.DEPT, donde:

- PERSON es el esquema de origen de datos
- DEPT es el nombre de tabla de origen de datos

La sentencia `SELECT * FROM NFXDEPT` se admite del servidor federado. Sin embargo, la sentencia `SELECT * FROM PERSON.DEPT` no está permitida (excepto en una sesión de paso a través). El servidor federado no tiene registrado PERSON.DEPT como apodo.

Ejemplo: Utilización de apodos en la sentencia CREATE TABLE

Se desea crear una tabla local basada en una tabla remota para la que se ha definido un apodo. Un ejemplo de la sentencia `CREATE TABLE` es:

```
CREATE TABLE nombre_tabla LIKE apodo
```

Apodos en sentencias DDL

Los objetos de origen de datos deben tener apodos registrados en la base de datos federada antes de que puedan incluirse en las sentencias DDL. Este tema proporciona algunos ejemplos de sentencias DDL que se utilizan con sistemas federados.

Utilización de apodos en la sentencia COMMENT ON

La sentencia `COMMENT ON` añade o sustituye comentarios al catálogo global de base de datos federada. La sentencia `COMMENT ON` es válida con un apodo y columnas definidas en un apodo. Esta sentencia no actualiza catálogos de origen de datos.

Utilización de apodos en las sentencias GRANT y REVOKE

Las sentencias `GRANT` y `REVOKE` son válidas con un apodo para determinados privilegios y para todos los usuarios y grupos. Sin embargo, el sistema federado no emite una sentencia `GRANT` o `REVOKE` correspondiente sobre el objeto del origen de datos a la que hace referencia el apodo.

Por ejemplo, suponga que el usuario JON crea un apodo para una tabla Oracle que no tenía índice. El apodo es ORAREM1. Posteriormente, el DBA de Oracle define un índice para esta tabla. Ahora la usuario EILEEN desea que la base de datos federada sepa que este índice existe, para que el optimizador de consultas pueda diseñar estrategias para acceder a la tabla de modo más eficaz. EILEEN puede informar a la base de datos federada que existe un índice nuevo, creando una especificación de índice para ORAREM1.

La información sobre el índice está almacenada en la vista de catálogos de SYSSTAT.INDEXES. Utilice la sentencia GRANT para dar a EILEEN el privilegio de índice sobre este apodo, de modo que pueda crear la especificación de índice.

```
GRANT INDEX ON NICKNAME ORAREM1 TO USER EILEEN
```

Para revocar los privilegios del usuario EILEEN para crear una especificación de índice en el apodo ORAREM1, utilice la sentencia REVOKE:

```
REVOKE INDEX ON ORAREM1 FROM USER EILEEN
```

Impacto de las estadísticas del origen de datos en las aplicaciones

Cuando se crea un apodo para un objeto de origen de datos, el catálogo global de la base de datos federada se actualizará con información sobre dicho objeto. El optimizador de consultas utiliza esta información para planificar el modo de recuperar datos del objeto.

Es importante asegurarse de que la información del origen de datos sea actual. La base de datos federada no detecta cambios automáticamente en objetos de origen de datos.

Estadísticas de objeto de base de datos almacenadas en el catálogo global

La información almacenada en el catálogo global acerca de un objeto de origen de datos, depende del tipo de objeto. Para las vistas y tablas de base de datos, el nombre del objeto, los nombres de columna y los atributos, se almacenan en el catálogo global.

En el caso de una tabla o apodo, la información también incluye:

- Estadísticas. Por ejemplo, el número de filas y el número de páginas en los que existe la fila. Para asegurarse de que la base de datos federada obtenga las últimas estadísticas, ejecute el origen de datos equivalente del mandato RUNSTATS en la tabla antes de crear el apodo.
- Descripciones de índice. Si la tabla no tiene índices, puede proporcionar al catálogo los metadatos que contiene normalmente una definición de índice. Por ejemplo, suponga que se crea un apodo para una tabla remota y que a continuación, se crea un índice en la tabla en el origen de datos. Puede crear una especificación de índice en el servidor federado que representa este índice remoto. Una especificación de índice se crea emitiendo la sentencia CREATE INDEX y haciendo referencia al apodo para la tabla. La cláusula SPECIFICATION ONLY se utiliza con la sentencia CREATE INDEX únicamente para producir una especificación de índice. La especificación de índice informa al optimizador federado que existe un índice remoto. Sin embargo, sólo se generan los metadatos. En realidad, en el servidor federado no se crea ningún índice. Además, no se proporciona ninguna información estadística al catálogo global. Si a la especificación de índice le proporciona exactamente la misma signatura que al índice remoto (es decir, el mismo nombre y las mismas

columnas en el mismo orden), podrá utilizar SYSPROC.NNSTAT para actualizar estadísticas en la especificación de índice y apodo.

Para determinar la información de origen de datos que se almacena en el catálogo global, consulte las vistas de catálogo SYSCAT.TABLES y SYSCAT.COLUMNS. Para determinar la información de índice de origen de datos que se almacena en el catálogo global o la especificación de índice concreta que contiene, consulte la vista de catálogo SYSCAT.INDEXES.

Actualización de estadísticas utilizando la vista SYSSTAT en vez de la vista SYSCAT

Las vistas SYSCAT son vistas de catálogo de sólo lectura en el esquema de SYSCAT. Las vistas de SYSSTAT son vistas de catálogo actualizables que contienen información estadística que utiliza el optimizador. Las vistas de SYSSTAT están en el esquema de SYSSTAT.

Si emite una operación UPDATE o INSERT en una vista del esquema de SYSCAT, la operación fallará. Utilice las vistas de catálogo actualizables del esquema de SYSSTAT para modificar manualmente las estadísticas sobre apodos.

Definición de opciones de columna en apodos

Las opciones de columna son parámetros de las sentencias CREATE NICKNAME y ALTER NICKNAME. Puede especificar opciones de columna al crear inicialmente un apodo o al modificar un apodo existente.

La información que facilite por medio de las opciones de columna se almacenará en el catálogo global.

Orígenes de datos no relacionales

Las opciones de columna son exclusivas para cada derivador no relacional. Estas opciones se establecen habitualmente al emitir la sentencia CREATE NICKNAME.

Orígenes de datos relacionales

Hay dos opciones de columna que pueden utilizarse para los orígenes de datos relacionales: NUMERIC_STRING y VARCHAR_NO_TRAILING_BLANKS.

Cómo establecer la opción de columna NUMERIC_STRING

En el caso de que una columna de serie de origen de datos sólo contenga dígitos numéricos y ningún otro carácter, incluyendo los blancos, establezca la opción de columna NUMERIC_STRING en Y.

Establecer la opción de columna NUMERIC_STRING en Y permite que las consultas que utilizan esta columna se optimicen para clasificar operaciones y operaciones de comparación. Por ejemplo:

```
ALTER NICKNAME nickname
  ALTER COLUMN local_column_name
  OPTIONS (SET NUMERIC_STRING 'Y')
```

Cómo establecer la opción de columna VARCHAR_NO_TRAILING_BLANKS

Si la columna de serie de origen de datos no contiene blancos de cola, establezca la opción de columna de VARCHAR_NO_TRAILING_BLANKS en Y.

Algunos orígenes de datos, por ejemplo, Oracle, no utilizan la misma lógica de comparación de serie rellena con blancos que utiliza la base de datos federada. Esto se aplica a los tipos de datos como por ejemplo, VARCHAR y VARCHAR2. En consecuencia, el optimizador de consultas debe volver a grabar predicados que impliquen estos tipos de datos para asegurar resultados de consulta coherentes.

Volver a grabar sentencias de consulta puede afectar al rendimiento. Establecer esta opción para una determinada columna proporciona al optimizador de columnas información sobre dichas columnas de modo que pueda generar sentencias SQL más eficaces.

Por ejemplo:

```
ALTER NICKNAME nickname
ALTER COLUMN local_column_name
OPTIONS (SET VARCHAR_NO_TRAILING_BLANKS 'Y')
```

Capítulo 18. Creación y utilización de vistas federadas

Una vista que incluya una referencia a un apodo en la selección completa es una *vista federada*. Se hace referencia a las tablas base en la vista federada utilizando apodos, en vez de utilizando los nombres de tabla de origen de datos.

Restricciones

Las vistas federadas que se crean a partir de varios objetos de origen de datos son vistas de sólo lectura y no pueden actualizarse.

Es posible que las vistas federadas que se crean a partir de un único objeto de origen de datos sean o no vistas de sólo lectura.

- Una vista federada creada a partir de un único origen de datos no relacional es de sólo lectura.
- Es posible que una vista federada creada a partir de un único origen de datos relacional permita actualizaciones, en función de lo que se incluye en la sentencia CREATE VIEW.

Acerca de esta tarea

Las ventajas de la utilización de vistas federadas son similares a las ventajas de utilizar vistas definidas en las tablas locales en un gestor de base de datos relacional centralizada:

- Las vistas proporcionan una representación integrada de los datos
- Puede excluir de una vista las columnas de tabla que contengan datos sensibles o confidenciales

Procedimiento

Una vista federada se crea a partir de los objetos de origen de datos que tengan apodos. La acción de crear una vista de base de datos federada de datos de origen de datos a veces se denomina “creación de una vista en un apodo”. Esta frase refleja el hecho de que para la vista federada que ha de crearse, la selección completa de la sentencia CREATE VIEW debe hacer referencia al apodo de cada una de las vistas y tablas de origen de datos que vaya a contener la vista federada.

Creación de vistas federadas - ejemplos

Este tema proporciona ejemplos de creación de vistas federadas

Ejemplo: creación de una vista federada que fusiona datos similares procedentes de varios objetos de origen de datos

Se está trabajando con datos de cliente de tres servidores independientes, uno en Europa, otro en Asia y un tercero en Sudamérica. Los datos del cliente de Europa están en una tabla de Oracle. El apodo para dicha tabla es ORA_EU_CUST. Los datos del cliente de Asia están en una tabla de Sybase. El apodo para dicha tabla es SYB_AS_CUST. Los datos del cliente de Sudamérica residen en una tabla de Informix. El apodo para dicha tabla es INFMX_SA_CUST. Cada tabla tiene columnas que contienen el número del cliente (CUST_NO), el nombre del cliente

(CUST_NAME), el número del producto (PROD_NO) y la cantidad pedida (QUANTITY). La sintaxis para crear una vista a partir de estos tres apodos que fusiona estos datos de cliente es:

```
CREATE VIEW FV1
  AS SELECT * FROM ORA_EU_CUST
  UNION
  SELECT * FROM SYB_AS_CUST
  UNION
  SELECT * FROM INFMX_SA_CUST
```

Ejemplo: unión de datos para crear una vista federada

Suponga que trabaja con datos de cliente situados en un servidor y datos de ventas situados en otro servidor. Los datos de cliente están en una tabla Oracle. El apodo para dicha tabla es ORA_EU_CUST. Los datos de ventas están en una tabla de Sybase. El apodo para dicha tabla es SYB_AS_SALES. Se desea emparejar la información de cliente con las compras efectuadas por dichos clientes. Cada tabla tiene una columna que contiene el número de cliente (CUST_NO). La sintaxis para crear una vista federada a partir de estos dos apodos que une estos datos es:

```
CREATE VIEW FV4
  AS SELECT A.CUST_NO, A.CUST_NAME, B.PROD_NO, B.QUANTITY
  FROM ORA_EU_CUST A, SYB_SALÉS B
  WHERE A.CUST_NO=B.CUST_NO
```

Capítulo 19. Mantener la integridad de los datos con niveles de aislamiento

El nivel de aislamiento define el grado de aislamiento para un proceso de aplicación a partir de los demás procesos de aplicación que se estén ejecutando simultáneamente.

Puede mantener la integridad de los datos para una tabla de origen de datos solicitando que las filas de las tablas se bloqueen en un determinado nivel de aislamiento.

El bloqueo se produce en la fila de la tabla base en el origen de datos. Sin embargo, el gestor de la base de datos puede sustituir varios bloqueos de filas por un único bloqueo de tabla. Esta acción se denomina *escala de bloqueos*. A un proceso de aplicación se le garantiza como mínimo el nivel de bloqueo mínimo que se haya solicitado.

Los niveles de aislamiento para la base de datos federada son como sigue:

- RR** Lectura repetible
- RS** Estabilidad de lectura
- CS** Estabilidad de cursor (valor por omisión)
- UR** Lectura no confirmada

Los tipos de aislamiento son el aislamiento a nivel de sentencia y el aislamiento a nivel de conexión.

Puede establecer el aislamiento cuando se realizan las acciones siguientes:

- Precompilar o vincular una aplicación. Puede especificar niveles de aislamiento al preparar o vincular una aplicación. El nivel de aislamiento especificado en el mandato BIND y PREP es el nivel de aislamiento por omisión cuando el servidor federado se conecta al origen de datos remoto.
- Utilizar la cláusula WITH en una sentencia SQL. Esta acción se denomina aislamiento a nivel de sentencia. Puede utilizar la cláusula WITH en las sentencias SELECT, UPDATE, INSERT y DELETE.

Si el servidor federado no encuentra un nivel de aislamiento para una sentencia, el servidor federado utilizará el nivel de aislamiento establecido cuando el servidor federado se conectó al origen de datos.

La tabla siguiente lista los orígenes de datos que utilizan el aislamiento a nivel de conexión, los niveles de aislamiento que utilizan y los niveles de aislamiento equivalentes en el servidor federado.

Tabla 21. Orígenes de datos y niveles de aislamiento

Orígenes de datos	El nivel de aislamiento más restrictivo	Nivel de aislamiento más restrictivo	Nivel de aislamiento menos restrictivo	El nivel de aislamiento menos restrictivo
Base de datos federada	Lectura repetible	Estabilidad de lectura	Estabilidad del cursor	Lectura no confirmada

Tabla 21. Orígenes de datos y niveles de aislamiento (continuación)

Orígenes de datos	El nivel de aislamiento más restrictivo	Nivel de aislamiento más restrictivo	Nivel de aislamiento menos restrictivo	El nivel de aislamiento menos restrictivo
Familia de productos de DB2	Lectura repetible	Estabilidad de lectura*	Estabilidad del cursor	Lectura no confirmada
Informix	Lectura repetible	Lectura repetible	Estabilidad del cursor	Lectura sucia
JDBC	Serializable	Lectura repetible	Lectura confirmada	Lectura no confirmada
Microsoft SQL Server	Serializable	Lectura repetible	Lectura confirmada	Lectura no confirmada
ODBC	Serializable	Lectura repetible	Lectura confirmada	Lectura no confirmada
Oracle	Serializable	Serializable	Lectura confirmada	Lectura confirmada
Sybase	Nivel 3	Nivel 3	Nivel 1	Nivel 0

*Para orígenes de datos DB2 para VM y VSE Server, el nivel de aislamiento es la lectura repetible.

El servidor federado no utiliza el registro especial CURRENT ISOLATION cuando se conecta a un origen de datos.

Los orígenes de datos no relacionales no tienen el concepto de niveles de aislamiento. OLE DB y Teradata si que tienen el concepto de niveles de aislamiento pero no están soportados por el servidor federado. No hay ninguna correlación de nivel de aislamiento entre los niveles de aislamiento de la base de datos federada y OLE DB, Teradata y los orígenes de datos no relacionales.

Aislamiento de nivel de sentencia en un sistema federado

Para los orígenes de datos federados, deberá utilizar la cláusula de aislamiento WITH para especificar el aislamiento de una sentencia.

Deberá utilizar la cláusula de aislamiento WITH en su sentencia si desea utilizar el aislamiento de nivel de sentencia. Si utiliza atributos con la Interfaz de nivel de llamada (CLI) u otra API de aplicación para el aislamiento de nivel de sentencia, no afecta al aislamiento de sentencia.

Los orígenes de datos que dan soporte al aislamiento de nivel de sentencia en un sistema federado son la familia de productos de DB2 y servidor SQL de Microsoft. El aislamiento de sentencia se envía a orígenes de datos remotos para el servidor SQL y familia de productos de DB2.

Utilice la opción de servidor DB2_STATEMENT_ISOLATION para activar o desactivar el aislamiento de nivel de sentencia. Puede especificar esta opción en las sentencias CREATE SERVER y ALTER SERVER. La opción de servidor se establece automáticamente en Y.

Puede utilizar la cláusula de aislamiento WITH en estas sentencias:

```
SELECT
```

SELECT INTO
DELETE búsqueda
INSERT
UPDATE búsqueda
DECLARE CURSOR

Cláusula de petición de bloqueo

Puede utilizar la cláusula de petición de bloqueo en una sentencia SELECT o SELECT INTO. Los orígenes de datos federados que dan soporte a la cláusula de petición de bloqueo son DB2 para Linux, UNIX y Windows, y DB2 para z/OS.

Limitaciones al utilizar la cláusula WITH para establecer el nivel de aislamiento.

Las condiciones siguientes se aplican a los niveles de aislamiento que se especifican para las sentencias:

- La cláusula WITH no puede utilizarse en las subconsultas.
- El nivel de aislamiento UR sólo se aplica si la tabla de resultados de la sentencia SELECT INTO de selección completa es de sólo lectura. En otros casos, el nivel de aislamiento de UR para la sentencia cambia de UR a CS para la familia de orígenes de datos de DB2. Para el origen de datos del servidor de SQL, el nivel de aislamiento de UR se actualiza a Lectura confirmada.
- Si especifica la cláusula de petición de bloqueo para los siguientes orígenes de datos, el servidor federado ignorará la cláusula.
 - DB2 para System i
 - DB2 para VM
 - Microsoft SQL Server

Aislamiento de nivel de conexión en un sistema federado

El servidor federado correlaciona su nivel de aislamiento con otro que se le corresponda en el origen de datos.

Para cada conexión con el origen de datos, el derivador determinará el nivel de aislamiento.

Cuando el servidor federado se conecta al origen de datos, el nivel de aislamiento del origen de datos remoto se establece en un nivel que equivale al nivel del servidor federado. Si no hay un equivalente exacto, el servidor federado establece el nivel de aislamiento en el nivel menos restrictivo. Una vez que se haya efectuado una conexión con un origen de datos, no podrá cambiarse el nivel de aislamiento mientras dure la conexión.

Todos los derivadores excepto Teradata siguen la pista al nivel de aislamiento de la conexión. Al configurar una conexión, los derivadores establecen el nivel de aislamiento de la conexión en el nivel equivalente al nivel de aislamiento actual. El nivel de aislamiento actual es el nivel de aislamiento de la sección actual (primera sentencia federada para un origen de datos). El derivador de Teradata siempre está en el nivel de aislamiento de READ, el valor por omisión, ya que el derivador de Teradata no tiene ninguna forma de cambiar el nivel de aislamiento actual.

Capítulo 20. Soporte de LOB federados

Con un sistema de base de datos federada, podrá acceder y manipular objetos grandes (LOB) en orígenes de datos remotos.

Un sistema federado da soporte a las operaciones SELECT en los LOB en orígenes de datos DRDA, Informix, Microsoft SQL Server, Oracle y Sybase. Por ejemplo:

```
SELECT empname, picture FROM infmx_emp_table
WHERE empno = '01192345'
```

Donde *picture* representa una columna LOB e *infmx_emp_table* representa un apodo que hace referencia a una tabla Informix que contiene datos de empleado.

Un sistema federado da soporte a las operaciones SELECT, INSERT, UPDATE y DELETE en los LOB en los siguientes orígenes de datos, utilizando el derivador DRDA:

- DB2 para z/OS (Versión 7 o posterior)
- DB2 para System i (versión 5)
- DB2 Database para Linux, UNIX y Windows (Versión 7 o posterior)

En la tabla siguiente se indican las operaciones de lectura y grabación a las que da soporte DB2 Database para Linux, UNIX y Windows:

Tabla 22. Soporte de lectura y grabación para los LOB

Origen de datos	Tipo de operaciones
DB2 para z/OS, DB2 para System i, DB2 Database para Linux, UNIX y Windows ¹	lectura y grabación
BioRS	sólo lectura
Informix	sólo lectura
JDBC	sólo lectura
Microsoft SQL Server	sólo lectura
Oracle (derivador de NET8) ²	lectura y grabación
ODBC	sólo lectura
Sybase	sólo lectura
Teradata	sólo lectura
Servicios Web	sólo lectura y salida de vinculación sólo para CLOB
XML	sólo lectura

Nota:

1. Se necesita DB2 para System i (versión 5 o posterior) para el soporte de LOB. DB2 Information Integrator Versión 8 no puede acceder a datos de tipo LOB de DB2 Database para Linux, UNIX y Windows Versión 7.
2. Para ejecutar operaciones de inserción, actualización y supresión en columnas LONG de Oracle, tendrá que migrar las columnas remotas de los LONG a los LOB y volver a crear los apodos.

LOB de Teradata

Los LOB de Teradata son ligeramente diferentes de los LOB de DB2.

Teradata no tiene ningún tipo de datos tan grande como el de los LOB soportados en el software DB2 para Linux, UNIX y Windows. Sin embargo, hay algunos tipos de datos de Teradata que pueden tener una longitud de hasta 64000 bytes. Estos tipos de datos son CHAR, VARCHAR, BYTE, VARBYTE, GRAPHIC y VARGRAPHIC. Estos tipos de datos de Teradata están correlacionados con los tipos de datos LOB de DB2 cuando la longitud del tipo de datos de Teradata supera los límites del tipo de datos de DB2 correspondiente.

Longitudes de LOB

Algunos orígenes de datos como, por ejemplo, Oracle e Informix no almacenan las longitudes de las columnas LOB en los catálogos de sus sistemas. Al crear un apodo en una tabla, se recupera la información procedente del catálogo del sistema de origen de datos, incluyendo la longitud de columna. Puesto que no existe ninguna longitud para las columnas LOB, la base de datos federada asume que dicha longitud es la longitud máxima de una columna LOB en DB2 Database para Linux, UNIX y Windows. La base de datos federada almacena la longitud máxima en el catálogo de la base de datos federada como la longitud de la columna de apodo.

Localizadores de LOB

Las aplicaciones pueden solicitar localizadores de LOB para los LOB almacenados en orígenes de datos remotos. Un localizador de LOB es un valor de 4 bytes almacenado en una variable de sistema principal. Una aplicación puede utilizar el localizador de LOB para hacer referencia a un valor de LOB (o expresión de LOB) que contenga el sistema de bases de datos.

Utilizando un localizador de LOB, una aplicación puede manipular el valor de LOB como si el valor de LOB se hubiera almacenado en una variable de sistema principal regular. Al utilizar localizadores de LOB, no es necesario transportar el valor de LOB del servidor de orígenes de datos a la aplicación (y, posiblemente, de la aplicación al servidor de orígenes de datos).

La base de datos federada puede recuperar los LOB de los orígenes de datos remotos, almacenarlas en el servidor federado y después emitir un localizador de LOB en el LOB almacenado. Los localizadores de LOB se liberan cuando:

- Las aplicaciones emiten sentencias FREE LOCATOR SQL
- Las aplicaciones emiten sentencias COMMIT
- La instancia federada se reinicia

Restricciones sobre los LOB

Los sistemas federados imponen algunas restricciones sobre los LOB.

A los LOB se les aplican las siguientes restricciones:

- La base de datos federada no puede vincular los LOB remotos con una variable de referencia de archivos
- Los LOB no están soportados en sesiones de paso a través
- Los LOB no se admiten como parámetros de procedimientos almacenados

Consideraciones de rendimiento para proceso de LOB

Al desarrollar aplicaciones federadas que captan y procesan datos LOB, los diseñadores de aplicaciones y administradores de base de datos han de comprender el modo en que el proceso de LOB afecta al rendimiento.

Cuando una aplicación capta datos de un origen de datos federado, el servidor federado debe captar los datos en sus propios almacenamientos intermedios de la aplicación antes de enviar los datos a la aplicación. Puesto que los LOB no se procesan en una agrupación de almacenamientos intermedios, los datos LOB deben pasar primero a través de un espacio de tabla temporal definido para el servidor federado. Para ayudar a mejorar el rendimiento y reducir el consumo de recursos, los diseñadores de aplicaciones sólo deberían materializar los datos de LOB cuando sea necesario.

De modo análogo, cuando el servidor federado actualice datos remotos, los datos deberán pasar a través de un espacio de tabla temporal asignado al servidor federado antes de pasarlos al origen de datos.

Los LOB transitorios utilizan el espacio de tabla temporal asignado al servidor federado. Por tanto, es posible que los administradores de base de datos tengan que aumentar el tamaño de este espacio de tabla temporal para asegurarse de que el área de trabajo es suficiente para procesar los LOB.

Recomendación: Para maximizar el rendimiento al trabajar con los LOB, defina el espacio de tabla temporal como sistema gestionado (SMS) y asegúrese de que el espacio de tabla temporal está ubicado en discos que tengan un ancho de banda de E/S alto.

Utilización de Interfaz de nivel de llamada de DB2 para acceder a los LOB federados

El servidor federado da soporte a dos API de DB2 CLI para seleccionar datos LOB:

- La API de SQLFetch capta el LOB del servidor o el origen de datos federado en los almacenamientos intermedios de la aplicación en una sola operación.
- La API de SQLGetData capta el LOB por partes y eso puede hacer que se necesiten sucesivas llamadas a la API para captar todo el LOB en los almacenamientos intermedios de la aplicación.

Recomendación: Para un óptimo rendimiento, utilice la API de SQLGetData al captar los LOB por medio de un servidor federado.

El servidor federado da soporte a las API de SQLExecute y SQLPutData para actualizar datos de LOB. La API de SQLExecute actualiza los datos de LOB en una única operación, en tanto que la API de SQLPutData puede necesitar sucesivas llamadas para enviar todos los datos de LOB de los almacenamientos intermedios de la aplicación al servidor. Cada API se ejecuta al mismo nivel en un entorno federado.

Derivadores fiables y protegidos

Los apodos creados para derivadores definidos como fiables o protegidos se ejecutan del mismo modo al captar o actualizar los LOB.

Capítulo 21. Peticiones distribuidas para consultar orígenes de datos

Las consultas que se envíen a la base de datos federada pueden solicitar resultados de un único origen de datos, pero normalmente son peticiones que incluyen varios orígenes de datos. Debido a que una consulta típica se distribuye a varios orígenes de datos, recibe el nombre de *petición distribuida*.

En general, una petición distribuida utiliza uno o más convenios SQL para especificar el lugar en el que los datos se van a recuperar de las subconsultas, establecer operadores y unir subselecciones.

Peticiones distribuidas para consultar orígenes de datos - ejemplos

Los ejemplos de este tema ilustran peticiones distribuidas con una subconsulta, operadores set y una operación join.

En los ejemplos siguientes, el servidor federado está configurado para acceder a un origen de datos DB2 para z/OS, un origen de datos DB2 para System i y un origen de datos Oracle. En cada origen de datos hay almacenada una tabla que contiene información de empleado. El servidor federado hace referencia a estas tablas mediante apodos que apuntan al lugar en el que residen las tablas.

zOS_EMPLOYEES

Apodo para una tabla de un origen de datos DB2 para z/OS que contenga información de empleado.

SYSTEMi_EMPLOYEES

Apodo para una tabla de un origen de datos DB2 para System i que contenga información de empleado.

ORA_EMPLOYEES

Apodo para una tabla de un origen de datos Oracle que contenga información de empleado.

ORA_REGIONS

Apodo para una tabla de un origen de datos Oracle que contenga información sobre las regiones en las que viven los empleados.

Los ejemplos siguientes ilustran los tres convenios de SQL utilizados con peticiones distribuidas, que utilizan los apodos definidos para cada una de las tablas.

Ejemplo: Petición distribuida con una subconsulta

SYSTEMi_EMPLOYEES contiene los números de teléfono de los empleados que viven en Asia. También contiene los códigos de región asociados con dichos números de teléfono, pero no lista las regiones que representan los códigos. ORA_REGIONS lista ambos códigos y regiones. La consulta siguiente utiliza una subconsulta para buscar el código de región para China. A continuación utiliza el código de región para devolver una lista de dichos empleados de SYSTEMi_EMPLOYEES que tengan un número de teléfono en China.

```
SELECT name,  
       telephone FROM  
       db2admin.SYSTEMi_employees
```

```

WHERE region_code IN
  (SELECT region_code FROM
dbadmin.ora_regions
  WHERE region_name = 'CHINA')

```

Ejemplo: Petición distribuida con operadores de conjunto

El servidor federado da soporte a tres operadores de conjunto: UNION, EXCEPT y INTERSECT.

- Utilice el operador de conjunto UNION para combinar las filas que satisfagan dos o más sentencias SELECT.
- Utilice el operador de conjunto EXCEPT para recuperar las filas que satisfagan la primera sentencia SELECT pero no la segunda.
- Utilice el operador de conjunto INTERSECT para recuperar las filas que satisfagan ambas sentencias SELECT.

Los tres operadores de conjunto pueden utilizar el operando ALL para indicar que no han de duplicarse filas del resultado. Lo cual elimina la necesidad de una clasificación adicional.

La consulta siguiente recupera todos los nombres de empleado y códigos de región presentes tanto en SYSTEMi_EMPLOYEES como en zOS_EMPLOYEES, aún en el caso de que cada una de las tablas resida en un origen de datos diferente.

```

SELECT name, region_code
  FROM as400_employees
INTERSECT
SELECT name, region_code
  FROM zOS_employees

```

Ejemplo: Petición distribuida para una unión

Una unión relacional produce un conjunto de resultados que contiene una combinación de columnas recuperadas de dos o más tablas. Deberían especificarse condiciones para limitar el tamaño de las filas en el conjunto de resultados.

La consulta que hay a continuación combina nombres de empleados y sus nombres de región correspondientes comparando los códigos de región listados en dos tablas. Cada una de las tablas reside en un origen de datos diferente.

```

SELECT t1.name, t2.region_name
  FROM dbadmin.SYSTEMi_employees t1, dbadmin.ora_regions t2
  WHERE t1.region_code = t2.region_code

```

Optimización de peticiones distribuidas con opciones de servidor

En un sistema federado, utilice parámetros denominados *opciones de servidor* para proporcionar al catálogo global información que se aplica a un origen de datos completo o a controlar el modo en que una base de datos interactúa con un origen de datos.

Acerca de esta tarea

Las opciones de servidor describen las posibilidades de un origen de datos concreto y mejora el conocimiento que el servidor federado tiene sobre dicho origen de datos. Por ejemplo, puede utilizar estas opciones de servidor:

- La opción de servidor VARCHAR_NO_TRAILING_BLANKS informa al optimizador que las columnas VARCHAR de entrada que residen en el servidor

de orígenes de datos están libres de blancos de cola. Utilice esta opción únicamente cuando esté seguro de que ninguna de las columnas VARCHAR2 para cada uno de los objetos a los que se hace referencia mediante un apodo en el servidor tiene blancos de cola. En caso contrario, utilice una opción de columna para especificar objetos individuales en el servidor que no tengan blancos de cola. La opción de columna también se denomina VARCHAR_NO_TRAILING_BLANKS.

- La opción de servidor PLAN_HINTS proporciona orígenes de datos Sybase con fragmentos de sentencia denominados *indicaciones de planes*. Las indicaciones de planes ayudan al optimizador de origen de datos a decidir el índice que ha de utilizarse al acceder a una tabla y la secuencia de unión de tablas que ha de utilizarse para recuperar datos para un conjunto de resultados.

Normalmente, el administrador de base de datos establece opciones de servidor para un sistema federado. Sin embargo, un programador puede hacer buen uso de las opciones de servidor que ayudan a optimizar consultas. Por ejemplo, para los orígenes de datos SYB1 y SYB2, la opción de servidor PLAN_HINTS se establece en el valor por omisión, N (no, no proporcionar indicaciones de planes a este origen de datos). Se escribe una petición distribuida que selecciona datos de SYB1 y SYB2. Se espera que los optimizadores de estos orígenes de datos puedan utilizar las indicaciones de planes para mejorar sus estrategias para acceder a estos datos. Puede alterar temporalmente el valor por omisión con un valor de Y (sí, proporcionar las indicaciones de planes) mientras la aplicación está conectada a la base de datos federada. Cuando termina la conexión a los orígenes de datos, el valor vuelve automáticamente a N.

Procedimiento

Para establecer opciones de servidor:

Utilice la sentencia SET SERVER OPTION para establecer o cambiar opciones de servidor. Para asegurarse de que el valor surte efecto, especifique la sentencia SET SERVER OPTION inmediatamente después de la sentencia CONNECT. La opción de servidor se establece por todo el tiempo de la conexión a la base de datos federada.

Tome en consideración la posibilidad de preparar la sentencia dinámicamente. La sentencia SET SERVER OPTION sólo afecta a las sentencias SQL dinámicas.

Para SQL estático, la sentencia SET SERVER OPTION sólo afecta a la ejecución de la sentencia SQL estática. El uso de la sentencia SET SERVER OPTION no afecta a los planes generados por el optimizador.

Cancelar una consulta federada

Puede interrumpir una aplicación y cancelar una consulta en ejecución. Con este soporte puede cancelar una consulta en la aplicación de orígenes de datos remoto antes de que finalice y propagar la interrupción y cancelación de la consulta al origen de datos remoto.

Restricciones

En la Versión 9.7, puede interrumpir una aplicación y cancelar una consulta en ejecución sólo en el origen de datos DB2 Database para Linux, UNIX y Windows.

Puede cancelar sentencia SELECT federada, sentencias INSERT, UPDATE y DELETE, procedimientos almacenados federados y sentencias en sesiones de paso a través.

Una sentencia federada puede contener múltiples segmentos que acceden a varios orígenes de datos. Si un segmento accede a un origen de datos que no está soportado, por ejemplo, múltiples segmentos de consulta contra el origen de datos Sybase, no podrá cancelar la sentencia.

No se puede cancelar una consulta federada cuando ATQ (cola de tablas asíncrona) está habilitada.

Acerca de esta tarea

Si el servidor federado está ejecutando una sentencia de SQL federada y está bloqueado esperando resultados y una respuesta del origen de datos remoto, la aplicación del servidor federado se encuentra en estado 'solicitud federada pendiente'. Puede emitir el mandato LIST APPLICATIONS para identificar aplicaciones que están en estado 'solicitud federada pendiente'.

Si una aplicación se encuentra en estado 'solicitud federada pendiente', puede utilizar Ctrl-C para interrumpir y cancelar una aplicación o el mandato FORCE APPLICATION para detener una aplicación en el servidor remoto.

Procedimiento

Para cancelar una consulta federada, utilice uno de los métodos siguientes:

- Pulse Ctrl-C para interrumpir la aplicación
Puede pulsar Ctrl-C para interrumpir la aplicación y cancelar la consulta que se está ejecutando en el servidor remoto. La sentencia de SQL actual del servidor federado se interrumpe y cancela, y se mantiene la coherencia de la base de datos local. La sentencia remota también se cancela. Las conexiones de salida y entrada permanecen activas.
- Emitir el mandato FORCE APPLICATION
Puede emitir el mandato FORCE APPLICATION para detener una aplicación. La aplicación se detiene en el servidor federado y se mantiene la coherencia de la transacción tanto localmente como en el servidor remoto. Las conexiones de entrada y salida se cierran y la parte remota de la ejecución de la aplicación en el origen de datos remoto, se cancela.

Capítulo 22. Consulta directa de orígenes de datos con paso a través

Utilice sesiones de paso a través para realizar operaciones que no son posibles con DB2 SQL/API.

Acerca de esta tarea

Las sesiones de paso a través son útiles cuando:

- Las aplicaciones deben crear objetos en el origen de datos o realizar operaciones INSERT, UPDATE o DELETE.
- La base de datos federada no da soporte a una operación de origen de datos exclusiva.

Procedimiento

Para consultar orígenes de datos directamente con paso a través:

- Utilice la sentencia SET PASSTHRU para iniciar una sesión de paso a través y acceder directamente a un servidor. Esta sentencia puede emitirse dinámicamente. Un ejemplo de esta sentencia es: SET PASSTHRU ORACLE1 Esta sentencia SET PASSTHRU abre una sesión de paso a través al origen de datos utilizando el nombre de servidor ORACLE1. ORACLE1 es el nombre que se registró para el servidor de orígenes de datos al crear la definición del servidor.
- Cuando se abra la sesión de paso a través, asegúrese de utilizar el nombre auténtico del objeto y no el apodo al hacer referencia a objetos en una sesión de paso a través. Deberá utilizar el dialecto de SQL del origen de datos, a menos que la base de datos federada sea el origen de datos al que se esté haciendo referencia.
- Si se envía una sentencia estática en una sesión de paso a través, ésta se enviará al servidor federado para su proceso. Si desea enviar una sentencia SQL a un origen de datos para su proceso, deberá prepararla dinámicamente en la sesión de paso a través y hacer que se ejecute mientras la sesión sigue abierta. Para preparar sentencias dinámicamente en una sesión de paso a través:
 - Para enviar una sentencia SELECT, utilice la sentencia PREPARE con la misma y después utilice las sentencias OPEN, FETCH y CLOSE para acceder a los resultados de la consulta.
 - Para una sentencia soportada que no sea SELECT, tendrá dos opciones. Podrá utilizar la sentencia PREPARE para preparar la sentencia soportada y después la sentencia EXECUTE para ejecutarla. Alternativamente, podrá utilizar la sentencia EXECUTE IMMEDIATE para preparar y ejecutar la sentencia.

Si emite el mandato COMMIT o ROLLBACK durante una sesión de paso a través, este mandato completará la unidad de trabajo actual, pero no finalizará la sesión de paso a través.

Consideraciones y restricciones de paso a través federado

Este tema explica las consideraciones y restricciones que ha de tener en cuenta al utilizar una sesión de paso a través.

Las siguientes consideraciones y restricciones se aplican a todos los orígenes de datos:

- Las sentencias preparadas en una sesión de paso a través deben ejecutarse en la misma sesión de paso a través. Las sentencias preparadas en una sesión de paso a través, pero ejecutadas fuera de la misma sesión de paso a través, fallarán y darán como resultado un error SQLSTATE 56098. Las sentencias preparadas fuera de una sesión de paso a través, pero ejecutadas dentro de la misma sesión de paso a través, se manejarán como una sentencia SET PASSTHRU.
- Aunque una aplicación puede emitir varias sentencias SET PASSTHRU, sólo la última sesión estará activa. Cuando se invoque una nueva sentencia SET PASSTHRU, ésta terminará la sentencia SET PASSTHRU anterior. No puede pasar a través de más de un origen de datos de la misma sesión de paso a través.
- Si se utilizan varias sesiones de paso a través en una aplicación, asegúrese de emitir un COMMIT antes de abrir otra sesión de paso a través. Esta acción concluirá la unidad de trabajo para la sesión actual.
- Los marcadores de parámetros no están soportados en sesiones de paso a través. Utilice variables de sistema principal en vez de marcadores de parámetro.
- Puede utilizar la semántica WITH HOLD en un cursor definido en una sesión de paso a través. No obstante, es posible que reciba un error si intenta utilizar la semántica WITH HOLD después de COMMIT y el origen de datos no da soporte a la semántica WITH HOLD.
- Las variables de sistema principal definidas en sentencias SQL en una sesión de paso a través deben adoptar el formato :Hn donde H está en mayúsculas y n es un número entero exclusivo. Los valores de n deben numerarse de modo consecutivo a partir de cero.
- Paso a través no da soporte a los LOB.
- Paso a través no da soporte a las llamadas de procedimiento almacenado.
- Paso a través no da soporte a la sentencia SELECT INTO.
- Paso a través no da soporte a las funciones definidas por el usuario externo o a SQL.
- No se pueden ejecutar sentencias SQL COMMIT o ROLLBACK dinámicas durante una sesión de paso a través.
- Al efectuar operaciones de actualización o supresión durante una sesión de paso a través, no podrá utilizar la condición WHERE CURRENT OF CURSOR.
- En la modalidad de paso a través, las sentencias SQL dinámicas se ejecutan remotamente, en tanto que las sentencias SQL estáticas se envían al servidor federado para su proceso. Si desea enviar una sentencia SQL a un origen de datos para su proceso, deberá prepararla dinámicamente en la sesión de paso a través y debe ejecutarse mientras la sesión siga abierta.
- Las sentencias SET PASSTHRU estáticas de procedimientos almacenados incorporados a SQL se bloquean cuando se crean procedimientos almacenados y se emite un error SQL0104N. Para entrar y salir de la modalidad de paso a través, utilice la sentencia EXECUTE IMMEDIATE.

Ejemplo:

```
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 curl;

end
DB20000I El mandato SQL se ha completado satisfactoriamente.

call stp()

```

```

Conjunto de resultados 1
-----
10 local
20 stp_insert

2 registro(s) seleccionado(s).

Estado de devolución = 0

```

```

select * from t1

C1 C2
-----
10 remote
20 passthru_insert

2 registro(s) seleccionado(s).

```

```

set passthru reset
DB20000I El mandato SQL se ha completado satisfactoriamente.

```

```

select * from t1

C1 C2
-----
10 local
20 stp_insert

2 registro(s) seleccionado(s).

```

- El retorno de un procedimiento almacenado o sentencia SQL compuesta no termina la modalidad de paso a través de modo automático. Para salir de la modalidad de paso a través, la sentencia SET PASSTHRU RESET deberá llamarse explícitamente, tal y como se muestra en el ejemplo anterior.

Sesiones de paso a través para orígenes de datos Oracle

Este tema identifica algunas consideraciones de SQL que han de tenerse en cuenta antes de someter sentencias SQL a orígenes de datos Oracle en una sesión de paso a través.

- Cualquier sentencia DDL emitida en un servidor Oracle se realizará en el momento del análisis y no estará sujeta a la semántica de transacciones. Cuando

se complete la operación, Oracle la confirmará automáticamente. Si se produce una retrotracción, el DDL no se retrotraerá.

- Cuando se emite una sentencia `SELECT` a partir de tipos de datos brutos, utilice la función `RAWTOHEX` para recibir los valores hexadecimales. Al realizar un `INSERT` en tipos de datos brutos, proporcione la representación hexadecimal.

Capítulo 23. Ajuste del proceso de consultas

Puede ajustar un sistema federado para mejorar el rendimiento del proceso de consultas.

Acerca de esta tarea

Cuando envía consultas SQL a la base de datos federada, el compilador SQL presiona la consulta y el optimizador de consultas la analiza y crea un plan de acceso. El optimizador de consultas almacena la información del plan de acceso en las tablas de Explain de la base de datos federada. Puede utilizar el formato de tabla de Explain y las herramientas db2expln y dynexpln para comprender el plan de acceso para una sentencia SQL determinada.

Procedimiento

Para ajustar los sistemas federados para mejorar el proceso de las consultas:

1. Envíe las consultas SQL que desee ajustar a la base de datos federada.
2. Analice dónde se evalúa la consulta.
3. Investigue los motivos de las decisiones que se han tomado en el plan de acceso y modifique el sistema para incrementar las oportunidades de envío.

Publicaciones sobre el rendimiento federado

Puede consultar muchos de los documentos de IBM que contienen información detallada sobre el ajuste del rendimiento.

- Asynchronous execution of federated queries in WebSphere Federation Server V9.1, en <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0611norwood/>
- Maximize the performance of WebSphere Information Integrator with Materialized Query Tables, en <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0605lin/index.html>
- Performance enhancements in IBM WebSphere Federation Server V9.1, Part 1: Improve performance of federated queries with new WebSphere Federation Server capabilities, en <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert/>
- Performance enhancements in IBM WebSphere Federation Server V9.1, Part 2: Performance characteristics of new functionality in WebSphere Federation Server, en <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0612englert2/>
- Using data federation technology in IBM WebSphere Information Integrator: Data federation usage examples and performance tuning, en <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0507lin/>
- Parallelism in WebSphere Information Integrator V8.2, en <http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0502harris/>
- Data Federation with IBM DB2 Information Integrator V8.1, en <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247052.html?Open>
- Using the federated database technology of IBM DB2 Information Integrator, en <ftp://ftp.software.ibm.com/software/data/pubs/papers/iifed.pdf>

Análisis de consultas

Una parte importante del proceso de consultas es el análisis, que determina cómo ajustar la consulta para obtener un rendimiento óptimo.

Para obtener datos desde orígenes de datos, los clientes (usuarios y aplicaciones) envían consultas en SQL a la base de datos federada. A continuación, el compilador de SQL consulta información en el catálogo global y en el derivador de orígenes de datos para ayudarlo a procesar la consulta. Esto incluye información sobre la conexión al origen de datos, atributos de servidor, correlaciones, información de índice y estadísticas de apodo.

Como parte del proceso del compilador de SQL, el *optimizador de consultas* analiza una consulta. El compilador desarrolla estrategias alternativas, llamadas *planes de acceso*, para procesar la consulta. Los planes de acceso pueden llamar a la consulta para que:

- La procesen los orígenes de datos.
- La procese el servidor federado.
- La procesen en parte los orígenes de datos y en parte el servidor federado.

La base de datos federada evalúa los planes de acceso principalmente en base a la información sobre las prestaciones del origen de datos y los atributos de los datos. El derivador y el catálogo global contienen esta información. La base de datos federada descompone la consulta en segmentos que se llaman *fragmentos de consulta*. Por lo general, es más eficaz enviar un fragmento de consulta a un origen de datos, si ésta puede procesar el fragmento. Sin embargo, el optimizador de consultas tiene en cuenta otros factores, tales como:

- La cantidad de datos que se debe procesar.
- La velocidad de proceso del origen de datos.
- La cantidad de datos que devolverá el fragmento.
- El ancho de banda de las comunicaciones.

El análisis de envío sólo se realiza en orígenes de datos relacionales. Los orígenes de datos no relacionales utilizan el protocolo solicitud-respuesta-compensar.

La siguiente figura ilustra los pasos que el compilador de SQL realiza cuando procesa una consulta.

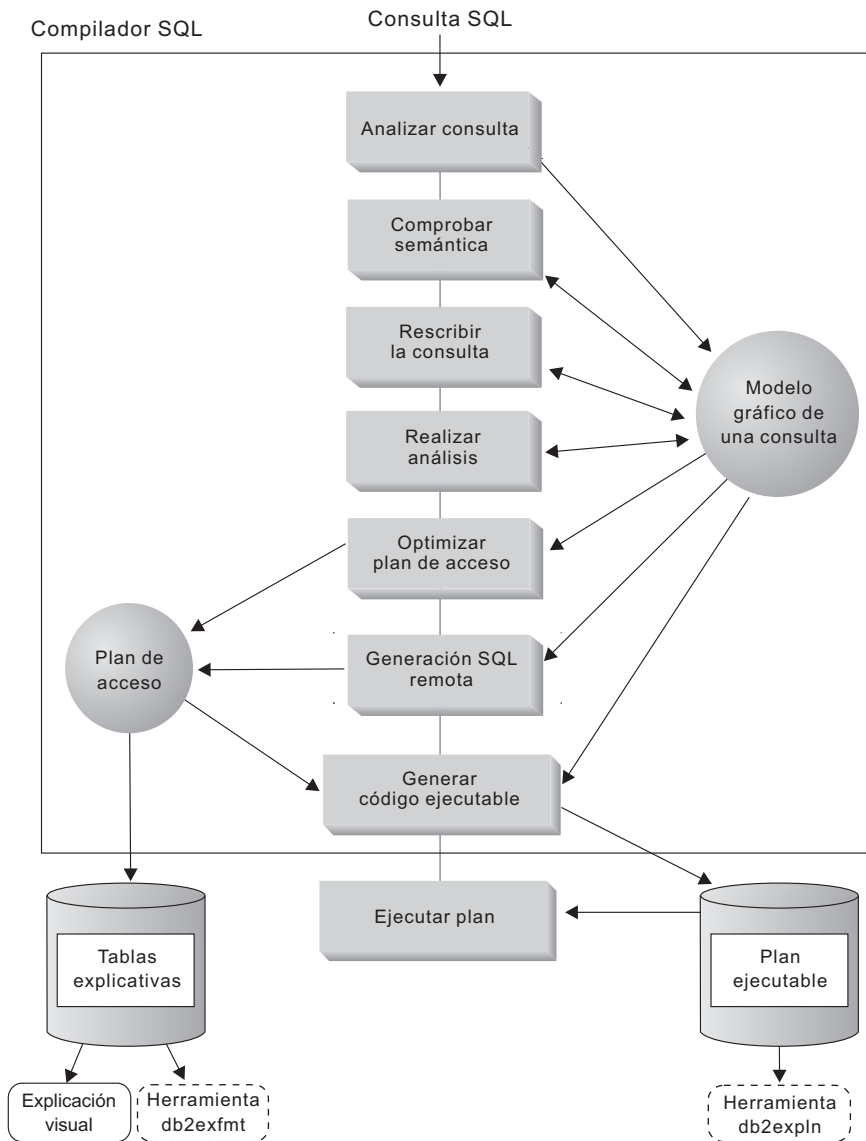


Figura 9. Diagrama de análisis de consultas del compilador de SQL

El optimizador de consultas genera planes de acceso locales y remotos para procesar un fragmento de consulta, en base al coste de recursos. A continuación, la base de datos federada elige el plan que considere que procesará la consulta con el menor coste de recursos.

Si alguna de los fragmentos se va a procesar mediante orígenes de datos, el servidor federado envía estos fragmentos a los orígenes de datos. Una vez que los orígenes de datos procesan los fragmentos, los resultados se recuperan y se devuelven al servidor federado. Si la base de datos federada ha realizado cualquier parte del proceso, éste combina sus resultados con los resultados recuperados desde el origen de datos. El servidor federado, a continuación, devuelve los resultados al cliente.

La tarea principal del análisis de envío es determinar qué operaciones se pueden evaluar remotamente. El análisis de envío realiza esta acción en base a la sentencia de SQL que reciba y en base a su conocimiento de las capacidades y de la semántica del origen de datos remoto. Basado en este análisis, el optimizador de

consultas evalúa las alternativas y elige el plan de acceso en función del coste. Puede que el optimizador elija no realizar una operación directamente en un origen de datos remoto debido a que sea menos rentable. Una segunda tarea es intentar volver a escribir la consulta para compensar la diferencia en semántica y en operaciones de SQL entre el servidor federado y el origen de datos, de manera que la consulta se optimice mejor.

El plan de acceso final seleccionado por el optimizador puede incluir operaciones evaluadas en los orígenes de datos remotos. Para aquellas operaciones que se realicen de manera remota, el compilador de SQL crea frases SQL eficaces en el dialecto de SQL del origen de datos remoto durante la fase de generación. El proceso de generar un plan de consultas óptimo que tenga en cuenta todos los orígenes se llama *optimización global*.

Para orígenes de datos no relacionales, los derivadores utilizan el protocolo solicitud-respuesta-compensar.

Análisis de envío

El análisis de envío indica al optimizador de consultas si un origen de datos remoto puede realizar una operación. Una operación puede ser una función, tal como un operador relacional, funciones del sistema o de usuario o un operador de SQL (GROUP BY, ORDER BY, etc.). A continuación, el optimizador toma una decisión basada en los costes acerca de si se debe enviar o no el operador. Incluso si el análisis de envío determina que se puede realizar una operación determinada en un origen remoto, es posible que el optimizador decida ejecutarla localmente en el servidor federado, si hacerlo así consume menos recursos.

El análisis de envío se realiza en orígenes de datos relacionales. Los orígenes no relacionales utilizan el protocolo petición-respuesta-compensar.

Las funciones que no se pueden enviar pueden afectar de forma significativa el rendimiento de la consulta. Tenga en cuenta el efecto de forzar un predicado selectivo para que sea evaluado localmente en lugar de en el origen de datos remoto. Este enfoque puede requerir que el servidor federado recupere toda la tabla del origen de datos remoto y, a continuación, filtre la tabla localmente utilizando el predicado. Si la red tiene restricciones y la tabla es grande, es posible que el rendimiento de la consulta resulte afectado.

Los operadores que no se envían también pueden afectar de forma significativa el rendimiento de la consulta. Por ejemplo, si un operador GROUP BY agrega datos remotos localmente, esto podría, una vez más, requerir que el servidor federado recuperara toda la tabla del origen de datos remoto.

Por ejemplo, suponga que el apodo EMP hace referencia a la tabla EMPLOYEE. Esta tabla tiene 10.000 filas. Una columna contiene los códigos de correo y otra columna contiene el salario para cada empleado. La consulta siguiente se envía al servidor federado para contar el número de empleados por ciudad que ganan más de 50.000 y que viven en un rango de códigos postales determinados:

```
SELECT CITY, COUNT(*) FROM EMP
WHERE ZIP BETWEEN 'CA1' AND 'CA5' AND SALARY > 50000
GROUP BY CITY;
```

Cuando el compilador de SQL recibe esta sentencia, considera varias posibilidades:

- Las secuencias de clasificación del origen de datos y el servidor federado son las mismas. Es probable que se envíen ambos predicados, porque es probable que reduzcan el tamaño del conjunto de resultados intermedio enviado desde el origen de datos al servidor federado. Normalmente es más eficaz filtrar y agrupar resultados en el origen de datos en lugar de copiar toda la tabla en el servidor federado y realizar las operaciones localmente. El análisis de envío determina si las operaciones se pueden realizar en el origen de datos. Debido a que las secuencias de clasificación son las mismas, los predicados y la operación GROUP BY pueden tener lugar en el origen de datos.
- Las secuencias de clasificación son las mismas y el optimizador de consultas sabe que el servidor federado es muy rápido. Es posible que el optimizador de consultas decida que la realización de la operación GROUP BY localmente es el mejor enfoque (menos costoso). Los predicados se enviarán al origen de datos para su evaluación. Se trata de un ejemplo de análisis de envío combinado con optimización global.
- Las secuencias de clasificación no son las mismas. Probablemente el predicado SALARY se enviará al origen de datos, porque las columnas numéricas se ordenan de la misma forma, independientemente de la secuencia de clasificación. Sin embargo, el predicado en ZIP no se enviará porque depende del orden en una columna de caracteres. GROUP BY no se enviará a menos que los predicados tanto en ZIP como en SALARY también se envíen.

El compilador de SQL considerará los planes de acceso disponibles y, a continuación, seleccionará el plan que sea más eficaz.

En general, el objetivo es garantizar que el optimizador de consultas considere enviar las funciones y los operadores a los orígenes de datos para su evaluación. Muchos factores pueden afectar si una función o un operador de SQL se evalúan en un origen de datos remoto. Los factores clave que influyen en el optimizador de consultas son: características del servidor, características del apodo y características de la consulta.

Características de servidor que afectan a oportunidades de envío

Las características de servidor que afectan al envío incluyen soporte de SQL, secuencia de clasificación, opciones de servidor federado y correlaciones de tipos y funciones.

Los factores que afectan a las oportunidades de envío para orígenes de datos no relacionales son distintos de los factores que afectan a las oportunidades de envío para orígenes de datos relacionales. El dialecto de SQL no es un factor para la mayoría de orígenes de datos no relacionales, puesto que no utilizan SQL.

Los siguientes temas describen los factores específicos de origen de datos que pueden afectar a las oportunidades de envío.

Diferencias de SQL

Las características de SQL que afectan al envío incluyen capacidades, restricciones, limitaciones de SQL, así como SQL específico de un servidor.

- Prestaciones de SQL. Cada origen de datos da soporte a una variante del dialecto de SQL y a distintos niveles de funcionalidad. Por ejemplo, considere la lista GROUP BY. La mayoría de orígenes de datos dan soporte al operador GROUP BY. Sin embargo, algunos orígenes de datos tienen restricciones sobre el número de elementos en la lista GROUP BY o restricciones sobre si se permite

una expresión en la lista GROUP BY. Si existe una restricción en el origen de datos remoto, es posible que el servidor federado realice la operación GROUP BY localmente.

- Restricciones de SQL. Cada origen de datos puede tener distintas restricciones de SQL. Por ejemplo, algunos orígenes de datos requieren marcadores de parámetros para vincular valores a sentencias de SQL remotas. Por lo tanto, se deben comprobar las restricciones de marcadores de parámetro para garantizar que cada origen de datos puede dar soporte a uno de estos mecanismos de vinculación. Si el servidor federado no puede determinar un buen método para vincular un valor para una función, esta función se debe evaluar localmente.
- Limitaciones de SQL. Es posible que el servidor federado permita la utilización de enteros más grandes que los orígenes de datos remotos. Sin embargo, los valores que exceden el límite no se pueden incorporar en sentencias que se envían a los orígenes de datos. Por lo tanto, la función o el operador que opera sobre esta constante se deben evaluar localmente.
- Detalles del servidor. Algunos factores se incluyen en esta categoría. Un ejemplo es la ordenación de valores NULL (el más alto, o el más bajo, en función de la ordenación). Por ejemplo, si el valor NULL se ordena en un origen de datos de forma distinta que en el servidor federado, las operaciones ORDER BY en una expresión con capacidad para nulos no se pueden evaluar de forma remota.

Tipo de datos VARCHAR2 en sistemas federados

Para ajustar el proceso de las consultas, el plan de acceso debe contabilizar el relleno con blancos con datos VARCHAR2.

La opción de servidor VARCHAR2_COMPAT se utiliza para habilitar el soporte para orígenes de datos compatibles con VARCHAR2.

La semántica de compatibilidad con VARCHAR2 determina la forma en que el servidor federado y el origen de datos manejan el tipo de datos VARCHAR2 en función de si el servidor federado, el origen de datos o ambos son compatibles con VARCHAR2. Las configuraciones posibles de los sistemas son las siguientes:

- El servidor federado es compatible con VARCHAR2, pero se conecta a un origen de datos que no lo es.
- El servidor federado no es compatible con VARCHAR2, pero se conecta a un origen de datos que sí lo es.
- Tanto el servidor federado como el origen de datos son compatibles con VARCHAR2.

Si el servidor federado o el origen de datos DB2 son compatibles con VARCHAR2, el tipo de datos VARCHAR2 se maneja como sinónimo del tipo de datos VARCHAR.

Consejo: Para garantizar una mejora en el rendimiento en sistemas que sólo se conectan a orígenes de datos compatibles con VARCHAR2, el servidor federado también debe ser compatible con VARCHAR2.

En los casos siguientes, el servidor federado maneja los valores de serie vacía y envía las operaciones al origen de datos en función de la semántica de compatibilidad con VARCHAR2.

Caso 1

El servidor federado no es compatible con VARCHAR2, pero se conecta a un origen de datos que sí lo es. El servidor federado permite series vacías, pero no así el origen de datos. Por tanto, todas las series vacías enviadas al origen de datos se convierten en valores de tipo NULL.

El comportamiento predeterminado del servidor federado es utilizar semántica de relleno con blancos en el tipo de datos CHAR antes de enviar el resultado del origen de datos. Sin embargo, el servidor federado no utiliza el relleno con blancos en operaciones que sólo especifican series vacías.

Ejemplos

- Las siguientes sentencias INSERT y UPDATE sólo incluyen series vacías y no se rellenan con blancos. Las sentencias envían valores de serie vacía que el origen de datos convierte en valores de tipo NULL.

Por ejemplo, el servidor federado no rellena con blancos las sentencias siguientes:

```
INSERT INTO n1 (c1) VALUES ('')
```

```
UPDATE n1 SET c1=''
```

```
INSERT INTO n1 (c1) VALUES (''), ('')
```

- La siguiente sentencia INSERT envía 10 blancos al origen de datos en lugar de la serie vacía:

```
INSERT INTO n1 (col_char) VALUES (''), ('ibm')
```

Caso 2

El servidor federado es compatible con VARCHAR2, pero se conecta a un origen de datos que no lo es. La semántica de compatibilidad con VARCHAR2 del servidor federado no permite series vacías ni tampoco utiliza semántica de relleno con blancos. Sin embargo, la semántica del origen de datos permite las series vacías y utiliza semántica de relleno con blancos. Por tanto, el servidor federado restringe el envío de las operaciones que incluyen series vacías, a menos que se conserven la semántica del servidor federado y la coherencia de los datos.

Ejemplos

- La siguiente sentencia INSERT se envía al origen de datos debido a que la serie vacía se convierte en un valor de tipo NULL y no puede manejarse localmente:

```
INSERT INTO n1 (c1) VALUES ('')
```

- La siguiente sentencia INSERT se envía al origen de datos si la sentencia se divide en dos sentencias separadas:

```
INSERT INTO n1 SELECT * FROM n2
```

Es necesario utilizar dos sentencias cuando las tablas del origen o del destino proceden de un origen de datos que no es compatible con VARCHAR2, por ejemplo:

```
SELECT * FROM n2  
INSERT INTO n1 VALUES (:H0)
```

- Las funciones que están anidadas dentro de expresiones se ejecutan localmente en el servidor federado; por ejemplo:

```
SELECT LENGTH(TRIM(c1)) FROM n1
```

- Todas las operaciones de comparación se ejecutan localmente en el servidor federado, ya que es posible que se generen resultados incoherentes si es el origen de datos el que ejecuta las operaciones de comparación.

Caso 3

Tanto el servidor federado como el origen de datos son compatibles con VARCHAR2. Ni el servidor federado ni el origen de datos permiten series vacías ni utilizan semántica de relleno con blancos. Por tanto, todas las operaciones que incluyen series vacías se envían al origen de datos, ya que la semántica se conserva.

Secuencia de clasificación

Si establece la opción del servidor `COLLATING_SEQUENCE` en 'Y', está indicando a la base de datos federada que la secuencia de clasificación del origen de datos coincide con la secuencia de clasificación del servidor federado. Este valor permite al optimizador considerar el proceso de envío dependiente del orden a un origen de datos, lo que puede mejorar el rendimiento.

Si la secuencia de clasificación del origen de datos no es la misma que la secuencia de clasificación de la base de datos federada y establece la opción del servidor `COLLATING_SEQUENCE` en 'Y', puede recibir resultados incorrectos. Por ejemplo, si el plan utiliza uniones de fusión, es posible que el optimizador envíe operaciones de ordenación a los orígenes de datos. Si la secuencia de clasificación del origen de datos no es la misma, es posible que la unión no tenga un conjunto de resultados correcto. Establezca la opción del servidor `COLLATING_SEQUENCE` en 'N', si no está seguro de si la secuencia de clasificación del origen de datos es idéntica a la secuencia de clasificación de la base de datos federada.

De forma alternativa, puede configurar una base de datos federada para utilizar la misma secuencia de clasificación que utiliza un origen de datos. A continuación, debe establecer la opción del servidor `COLLATING_SEQUENCE` en 'Y'. Esto permite al optimizador considerar el envío de operaciones dependientes del orden en columnas de caracteres.

Para determinar si un origen de datos y la base de datos federada tienen la misma secuencia de clasificación, tenga en cuenta los factores siguientes:

- Soporte de idioma nacional
La secuencia de clasificación está relacionada con el idioma soportado en un servidor. Compare la información de soporte de idioma nacional de la base de datos federada para el sistema operativo con la información de soporte de idioma nacional del origen de datos.
- Clasificaciones que tienen en cuenta el idioma
Compruebe si la base de datos federada o el origen de datos utilizan clasificaciones que tienen en cuenta el idioma. Si utilizan clasificaciones diferentes, no debe establecer `COLLATING_SEQUENCE` en Y.
- Características del origen de datos
Algunos orígenes de datos se crean utilizando secuencias de clasificación que no son sensibles a mayúsculas y minúsculas, que pueden proporcionar resultados distintos de la base de datos federada en operaciones dependientes del orden.
- Personalización

Algunos orígenes de datos proporcionan varias opciones para las secuencias de clasificación o permiten que se personalice la secuencia de clasificación.

Cuando un fragmento de consulta de un servidor federado requiere ordenación, el lugar donde se procesa la ordenación depende de varios factores. Si la secuencia de clasificación de la base de datos federada es la misma que la secuencia de clasificación del origen de datos, la ordenación puede tener lugar en el origen de datos o en el servidor federado. El optimizador de consultas puede determinar si una ordenación local o una ordenación remota es la forma más eficaz de completar la consulta.

Las comparaciones numéricas, en general, se pueden realizar en cualquiera de las dos ubicaciones incluso si la secuencia de clasificación es distinta. Sin embargo, puede obtener resultados incorrectos si el peso de los caracteres nulos es distinto entre la base de datos federada y el origen de datos.

De forma similar, para sentencias de comparación, tenga cuidado si está enviando sentencias a un origen de datos sensible a mayúsculas y minúsculas. Los pesos asignados a los caracteres "I" e "i" en un origen de datos que no distingue entre mayúsculas y minúsculas son iguales. Por ejemplo, en un origen de datos que no distinguiera entre mayúsculas y minúsculas con una página de códigos inglesa, **STEWART**, **SteWArT** y **stewart** se considerarían iguales. Por omisión, la base de datos federada es sensible a mayúsculas y minúsculas y asignaría pesos distintos a los caracteres.

Si las secuencias de clasificación de la base de datos federada y el origen de datos difieren, el servidor federado recupera los datos a la base de datos federada, de forma que puede realizar la ordenación localmente. La razón es que los usuarios esperan ver los resultados de la consulta ordenados según la secuencia de clasificación definida para el servidor federado; ordenando los datos localmente, el servidor federado garantiza que se cumple esta expectativa.

Si la consulta contiene un predicado de igualdad en la columna de caracteres, es posible enviar esa parte de la consulta incluso si las secuencias de clasificación son distintas (están establecidas en 'N'). Por ejemplo, el predicado `C1 = 'A'` recuperaría los mismos valores independientemente de la secuencia de clasificación y, por lo tanto, se podría enviar a un origen de datos que tuviera una secuencia de clasificación distinta que el servidor federado. Sin embargo, estos predicados no se pueden enviar cuando la secuencia de clasificación en el origen de datos no distingue entre mayúsculas y minúsculas (`COLLATING_SEQUENCE='I'`). Cuando un origen de datos no distingue entre mayúsculas y minúsculas, los resultados de `C1 = 'A'` y `C1 = 'a'` son los mismos, lo que no es coherente con un entorno sensible a mayúsculas y minúsculas tal como DB2 Database para Linux, UNIX y Windows.

Los administradores pueden crear bases de datos federadas con una secuencia de clasificación determinada que coincida con la secuencia de clasificación del origen de datos. Este enfoque puede acelerar el rendimiento si todos los orígenes de datos utilizan la misma secuencia de clasificación o si la mayoría o todas las funciones de columna se dirigen a orígenes de datos que utilizan la misma secuencia de clasificación. Por ejemplo, en DB2 para z/OS, las ordenaciones definidas por cláusulas `ORDER BY` son implementadas por una secuencia de clasificación basada en una página de códigos EBCDIC. Si desea utilizar el servidor federado para recuperar datos de DB2 para z/OS ordenados según cláusulas `ORDER BY`, es aconsejable configurar la base de datos federada de forma que ésta utilice una secuencia de clasificación predefinida en base a la página de códigos EBCDIC.

Si las secuencias de clasificación en la base de datos federada y en el origen de datos difieren, y necesita ver los datos ordenados en la secuencia del origen de datos, puede enviar la consulta en una sesión de paso a través o bien definir la consulta en una vista de origen de datos.

Opciones de servidor federado

Las opciones del servidor que establece acotan el conocimiento que tiene el servidor federado sobre el origen de datos remoto.

Los factores listados anteriormente que afectan a las oportunidades de envío son características de los servidores de bases de datos y no puede cambiarlos. Considerando cuidadosamente las siguientes opciones del servidor, es posible mejorar el rendimiento de la consulta:

- **COLLATING_SEQUENCE.** Si un origen de datos tiene una secuencia de clasificación que difiere de la secuencia de clasificación de la base de datos federada, no se puede evaluar de forma remota ninguna operación dependiente del orden sobre valores de caracteres en el origen de datos. Un ejemplo es la ejecución de funciones de columna MAX sobre una columna de caracteres de apodo en un origen de datos con una secuencia de clasificación distinta. Debido a que los resultados pueden diferir si se evalúa la función MAX en el origen de datos remoto, la base de datos federada realizará la función de agregación y la función MAX localmente.
- **VARCHAR_NO_TRAILING_BLANKS.** Esta opción es para series de caracteres de longitud variable que no contienen espacios en blanco al final. Algunos orígenes de datos, tal como Oracle, no aplican semántica de comparación con relleno de espacios en blanco como lo hace la base de datos federada. Esta diferencia de relleno puede causar resultados inesperados.

Por ejemplo:

```
'HELLO' = 'HELLO'      ' en DB2
'HELLO' <> 'HELLO'    ' en Oracle!
```

Si los espacios en blanco al final están presentes en columnas VARCHAR en un origen de datos Oracle, debe establecer esta opción en N (el valor por omisión para Oracle). Esta opción afecta al rendimiento, porque el servidor federado debe compensar la diferencia en semántica, pero garantiza un conjunto de resultados coherente. Si establece este valor en Y cuando una columna de origen de datos de Oracle contiene espacios en blanco al final puede causar resultados incoherentes.

Si está seguro de que las columnas VARCHAR y VARCHAR2 en un origen de datos no contienen espacios en blanco al final, considere establecer esta opción del servidor para un origen de datos. Asegúrese de considerar todos los objetos que pueden tener potencialmente apodos, incluyendo las vistas.

Recomendación: establezca esta opción individualmente para cada columna utilizando la opción de columna VARCHAR_NO_TRAILING_BLANKS.

- **DB2_MAXIMAL_PUSHDOWN.** Esta opción especifica los principales criterios que utiliza el optimizador de consultas al seleccionar un plan de acceso. El optimizador de consultas puede seleccionar planes de acceso en base al coste o en base al requisito del usuario de forma que se realiza el máximo proceso de consultas que sea posible por parte de los orígenes de datos remotos. Con DB2_MAXIMAL_PUSHDOWN establecido en Y, la reducción del tráfico de red se convierte en el criterio de alteración temporal para el optimizador de consultas. El optimizador de consultas utiliza el plan de acceso que realiza el número menor de "envíos" a los orígenes de datos. El establecimiento de esta opción del servidor en Y fuerza al servidor federado a utilizar un plan de acceso

que es posible que no sea el plan con el coste inferior. La utilización de un plan de acceso distinto del plan con el coste inferior puede disminuir el rendimiento. Cuando la opción del servidor `DB2_MAXIMAL_PUSHDOWN` se establece en `Y`, no se envía a los orígenes de datos remotos una consulta que resulta en un producto cartesiano. Las consultas que darán como resultado un producto cartesiano serán procesadas por la base de datos federada. La opción del servidor `DB2_MAXIMAL_PUSHDOWN` no necesita estar establecida en `Y` para que el servidor federado envíe proceso de consultas a los orígenes de datos remotos. Cuando esta opción del servidor se establece en `N` (valor por omisión), el optimizador de consultas enviará el proceso de consultas a los orígenes de datos. Sin embargo, los principales criterios que utiliza el optimizador cuando la opción se establece en `N` es el coste en lugar del tráfico de red.

“Características de servidor que afectan a la optimización global” en la página 271 describe las opciones del servidor `COMM_RATE`, `CPU_RATIO` y `IO_RATIO` que también pueden afectar al rendimiento de las consultas.

Factores de correlación de tipos y funciones

Tanto las correlaciones de tipos de datos por omisión como las correlaciones de funciones por omisión están incorporadas en los derivadores de origen de datos. Las correlaciones de tipos de datos describen la relación entre el tipo de datos del origen de datos y el tipo de datos del servidor federado. Puede personalizar las correlaciones de tipos de datos por omisión. Las correlaciones de funciones describen la relación entre una función de origen de datos y una función equivalente semánticamente en el servidor federado. En algunos casos, la base de datos federada compensará las correlaciones de funciones a las que no da soporte un origen de datos.

Las correlaciones de tipos de datos por omisión están diseñadas de forma que se proporcione suficiente espacio de almacenamiento intermedio a cada tipo de datos de origen de datos para evitar el truncamiento y desbordamiento de almacenamiento intermedio de tiempo de ejecución. Puede personalizar la correlación de tipos para un origen de datos específico o para un apodo específico a fin que se adapte a aplicaciones específicas y en algunos casos mejorar el rendimiento. Por ejemplo, los tipos `DATE` de Oracle pueden obtener tanto una parte de fecha como una parte de indicación de fecha y hora y, por lo tanto, están correlacionados con los `TIMESTAMP` de DB2 por omisión. Si está accediendo a una columna de fecha de Oracle y sabe que contiene sólo partes de fecha (no indicaciones de fecha y hora), puede utilizar la sentencia `ALTER NICKNAME` para cambiar el tipo de datos local del apodo de `TIMESTAMP` a `DATE` cuando la opción de servidor `date_compat` está desactivada. Cuando se evalúan predicados puramente en base a una fecha, tal como `SalesDate=DATE('2009-01-04')`, este cambio pasa por alto el uso de una función `ESCALAR` que se utiliza para extraer la fecha de la información de fecha y de indicación de fecha y hora contenida en la columna, lo que puede mejorar el rendimiento.

La base de datos federada compensa las funciones a las que no da soporte un origen de datos. La compensación funcional normalmente implica la recuperación de datos necesarios del origen de datos y la aplicación de la función de forma local, lo que a menudo tiene un impacto en el rendimiento. Existen varios casos en los que se produce una compensación de funciones:

- Una función simplemente no existe en el origen de datos. Algunas de las funciones de `SYSFUN`, por ejemplo, no existen en orígenes de datos de DB2 para `z/OS` y, por consiguiente, requieren compensación local.

- Una función existe en el origen de datos; sin embargo, las características del operando violan restricciones de función. Un ejemplo es el operador relacional IS NULL. La mayoría de orígenes de datos dan soporte al mismo, pero algunas tienen restricciones tal como sólo permitir un nombre de columna en el lateral izquierdo del operador IS NULL.
- Una función, si se evalúa remotamente, puede devolver un resultado distinto. Un ejemplo es el operador '>' (mayor que). Para aquellos orígenes de datos con secuencias de clasificación distintas, el operador mayor que puede devolver resultados distintos que si es evaluado localmente por la base de datos federada.

Características de apodo que afectan a oportunidades de envío

Las características de apodo que afectan al envío incluyen el tipo de datos local de una columna de apodo, las opciones de columna federada y las tablas de consultas materializadas.

Existen varios factores específicos del apodo que pueden afectar a las oportunidades de envío. El tipo de datos local de una columna de apodo puede afectar al número de posibilidades en una secuencia de unión evaluada por el optimizador. Los apodos pueden estar marcados con una opción de columna para indicar que las columnas no contienen blancos de cola. Este hecho proporciona al compilador de SQL la oportunidad de generar una forma de predicado más eficaz para la sentencia de SQL enviada a los orígenes de datos.

Tipo de datos locales de una columna de apodo

Asegúrese de que el tipo de datos local de una columna no impida que un predicado se evalúe en el origen de datos.

Las correlaciones de tipos de datos por omisión se proporcionan para evitar cualquier posible desbordamiento. Sin embargo, un predicado de unión entre dos columnas de longitudes o tipos de datos distintos impedirá que el optimizador considere una técnica de unión hash. Para que el optimizador considere la unión hash, tanto los tipos de datos como las longitudes de las columnas de la unión deben coincidir exactamente. Por ejemplo, las columnas de origen de datos Oracle diseñadas para albergar sólo valores enteros a menudo se crean como NUMBER en la base de datos Oracle, que adopta el valor por omisión de NUMBER (38). Una columna de apodo para este tipo de datos Oracle recibe el tipo de datos local FLOAT porque el rango de un entero de DB2 es sólo aproximadamente igual a NUMBER (9). En este caso, las uniones entre una columna de entero de DB2 y una columna de Oracle que está definida como NUMBER (pero sólo alberga valores enteros) no puede utilizar la técnica de unión hash porque la columna de Oracle está correlacionada como un tipo FLOAT. Sin embargo, si el dominio de esta columna NUMBER de Oracle puede ser acomodado por el tipo de datos INTEGER de DB2, puede cambiar su tipo de datos local con la sentencia ALTER NICKNAME. A continuación, el optimizador puede considerar la técnica de unión hash, que puede mejorar el rendimiento.

Opciones de columna federada

Puede definir opciones de columna federada que el optimizador de consultas utiliza para desarrollar planes de acceso.

Las opciones de columna le indican al derivador que maneje los datos de una columna de forma distinta a como normalmente los manejaría. El compilador de SQL y el optimizador de consultas utilizan los metadatos para desarrollar mejores planes para acceder a los datos. La base de datos federada trata el objeto al que

hace referencia un apodo como si fuese una tabla. Como resultado, el usuario puede definir opciones de columna para cualquier objeto de origen de datos para el que se cree un apodo.

La sentencia ALTER NICKNAME se puede utilizar para añadir o cambiar opciones de columna para apodos. Existen dos opciones de columna:

- **NUMERIC_STRING.** Esta opción de columna se aplica a columnas de tipo carácter (CHAR y VARCHAR). Suponga que un origen de datos tiene una secuencia de clasificación que difiere de la secuencia de clasificación de la base de datos federada. El servidor federado no clasificaría ninguna columna que contuviese datos de carácter en el origen de datos. Devolvería los datos a la base de datos federada y realizaría la clasificación localmente. Sin embargo, suponga que los datos de la columna son de tipo carácter y que la columna sólo contiene caracteres numéricos ('0','1',..., '9'). Puede indicar este hecho asignando un valor 'Y' a la opción de columna NUMERIC_STRING. Esto proporciona al optimizador de consultas la posibilidad de realizar la ordenación en el origen de datos puesto que los valores numéricos, incluso cuando se representan como series de caracteres, siempre se clasifican de la misma manera independientemente de la secuencia de clasificación. Si la ordenación se realiza de forma remota, puede evitar la actividad que supone transferir los datos al servidor federado y realizar la clasificación localmente.
- **VARCHAR_NO_TRAILING_BLANKS.** A diferencia de la opción del servidor con el mismo nombre, esta opción de columna se puede utilizar para identificar columnas de Oracle específicas que no contengan blancos de cola. El paso de análisis de envío del compilador de SQL tomará en consideración esta información al comprobar todas las operaciones realizadas en columnas que tengan este valor. En base al valor VARCHAR_NO_TRAILING_BLANKS, el compilador de SQL puede generar una forma de predicado diferente pero semánticamente equivalente que se utilice en la sentencia de SQL remota enviada al origen de datos. Es probable que un valor 'Y' habilite el uso de índices remotos (si están disponibles) que puedan mejorar el rendimiento de la consulta.

Características de consulta que afectan a oportunidades de envío

Un operador de SQL que hace referencia a varios orígenes de datos afecta al envío.

Una consulta puede hacer referencia a un operador de SQL que implique apodos de diversos orígenes de datos. Cuando el servidor federado combina los resultados de dos orígenes de datos referenciados utilizando un operador, la operación se debe llevar a cabo en el servidor federado. Un ejemplo de esto es un operador determinado, como UNION. El operador no se puede evaluar en un origen de datos remoto directamente.

Análisis del lugar donde se evalúa una consulta

La información detallada del optimizador de consultas se mantiene en tablas de Explain separada del propio plan de acceso real. Esta información permite un análisis en profundidad de un plan de acceso. Examinando el operador SHIP de un plan de acceso federado, puede determinar qué operaciones de SQL se han enviado a un origen de datos y qué operaciones se han ejecutado en el servidor federado.

Las tablas de Explain son accesibles en todos los sistemas operativos soportados y contienen información tanto para sentencias de SQL estáticas como dinámicas. Las herramientas siguientes se utilizan normalmente para obtener información de planes de acceso de las tablas de Explain:

- Herramienta de formato de tabla Explain. Utilice la herramienta db2exfmt para presentar la información de las tablas de Explain en un formato predefinido.
- Herramientas db2expln y dynexpln. Puede utilizar estas herramientas para comprender el plan de acceso elegido para una sentencia de SQL determinada. También puede utilizar el recurso de Explain integrado del Centro de control de DB2 conjuntamente con Visual Explain para comprender el plan de acceso elegido para una sentencia de SQL determinada. Tanto las sentencias de SQL estáticas como las sentencias de SQL dinámicas se pueden explicar utilizando el recurso de Explain. Una diferencia de las herramientas de Explain es que con Visual Explain la información de Explain se presenta en un formato gráfico. De lo contrario, el nivel de detalle proporcionado en los dos métodos es equivalente. Para utilizar completamente la salida de db2expln y dynexpln debe comprender lo siguiente:
 - Las distintas sentencias de SQL soportadas y la terminología relativa a estas sentencias (tal como predicados en una sentencia SELECT)
 - El propósito de un paquete (plan de acceso)
 - El propósito y el contenido de las tablas de catálogo del sistema
 - Conceptos generales del ajuste de la aplicación

También puede acceder a las tablas de Explain utilizando sentencias de SQL. Esto permite una fácil manipulación de la salida, para realizar comparaciones entre distintas consultas o para realizar comparaciones de la misma consulta a lo largo del tiempo.

Análisis del lugar donde se evalúa una consulta con la opción de servidor DB2_MAXIMAL_PUSHDOWN

Puede utilizar la opción del servidor DB2_MAXIMAL_PUSHDOWN conjuntamente con los programas de utilidad de Explain para determinar si se ha enviado un operador determinado para ejecutarse en un origen de datos debido a una decisión basada en costes del optimizador o porque el análisis de envío ha determinado que no era posible.

Procedimiento

Para ejecutar las herramientas de Explain en una consulta con la opción del servidor DB2_MAXIMAL_PUSHDOWN:

1. Establezca la opción del servidor DB2_MAXIMAL_PUSHDOWN en 'N'. Éste es el valor por omisión para esta opción. El análisis de envío determina qué partes del SQL se pueden enviar. A continuación, el optimizador de consultas genera todos los planes alternativos que no violan los criterios establecidos por el análisis de envío. El optimizador de consultas calcula el coste de cada plan y seleccionará el plan con el coste estimado más bajo. Puede analizar los operadores que se han enviado al origen de datos visualizando los detalles del operador SHIP adecuado. Si un operador que espera que se envíe no se ha enviado, continúe con el paso 2.
2. Establezca la opción del servidor DB2_MAXIMAL_PUSHDOWN en 'Y'. Utilice las herramientas de Explain para analizar de nuevo la sentencia de SQL. El plan visualizado en la salida de Explain visualiza todas las operaciones de SQL que se pueden enviar al origen de datos.

- Si el operador se envía después de restablecer la opción en 'Y', el optimizador ha determinado que era más económico ejecutar el operador localmente que hacerlo de forma remota. Si el operador no se envía después de restablecer la opción en 'Y', es probable que el análisis de envío no haya permitido que el operador se ejecute de forma remota.
- Si el optimizador ha realizado una decisión en base a los costes de no enviar el operador, considere comprobar las estadísticas de apodo para asegurarse de que son precisas. Si el análisis de apodo ha realizado la decisión de no enviar el operador, considere comprobar las opciones del servidor, las correlaciones de tipos de datos y las correlaciones de funciones.

Interpretación de decisiones de evaluación de planes de acceso

Los temas de esta sección listan preguntas típicas sobre el análisis de plan de acceso y áreas que puede investigar para aumentar las oportunidades de envío.

Por qué no se evalúa este predicado de forma remota

Esta cuestión surge cuando un predicado es muy selectivo y por lo tanto se podría utilizar para filtrar filas y reducir el tráfico de la red. La evaluación de predicado remoto también afecta a si una unión entre dos tablas del mismo origen de datos se puede evaluar de forma remota.

Las áreas a examinar incluyen:

- Opciones de servidor. ¿Cómo afectan los valores de las opciones del servidor COLLATING_SEQUENCE y VARCHAR_NO_TRAILING_BLANKS dónde se evalúa el predicado?
- Predicados de subconsulta. ¿Contiene este predicado una subconsulta que pertenezca a otro origen de datos? ¿Contiene este predicado una subconsulta que implique un operador de SQL que no esté soportado por este origen de datos? No todos los orígenes de datos dan soporte a operadores set en un predicado.
- Funciones de predicado. ¿Contiene este predicado una función que no puede ser evaluada por este origen de datos remoto? Los operadores relacionales se clasifican como funciones.
- Requisitos de vinculación de predicado. ¿Requiere este predicado, si se evalúa de forma remota, la vinculación de algún valor? Si es así, ¿violaría las restricciones de SQL en este origen de datos?
- Optimización global. El optimizador ha decidido que el proceso local es más económico.

Por qué el operador GROUP BY no se evalúa de forma remota

Existen varias áreas que puede comprobar para determinar porqué un operador GROUP BY no se evalúa de forma remota.

Las áreas que puede comprobar incluyen:

- ¿Se evalúa correctamente la entrada al operador GROUP BY? Si la respuesta es no, examine la entrada.
- ¿Tiene el origen de datos restricciones sobre este operador? Los ejemplos incluyen:
 - Número limitado de elementos GROUP BY
 - Recuentos limitados de bytes de elementos GROUP BY combinados
 - Especificación de columna sólo en la lista GROUP BY

- ¿El origen de datos da soporte a este operador de SQL?
- Optimización global. El optimizador ha decidido que el proceso local es más económico.

Por qué no se evalúa el operador SET de forma remota

Puede comprobar los operandos y comprobar las restricciones de origen de datos para determinar por qué el operador SET no se evalúa de forma remota.

Consideraciones:

- ¿Se evalúan estos dos operandos completamente en el mismo origen de datos remoto? Si la respuesta es no y debe ser sí, examine cada operando.
- ¿Tiene el origen de datos alguna restricción sobre este operador SET? Por ejemplo, ¿son los objetos grandes o los campos largos una entrada válida para este operador SET específico?

Por qué no se evalúa la operación ORDER BY de forma remota

Puede comprobar la entrada de la operación, lo que contiene la cláusula, y comprobar las restricciones de origen de datos para determinar por qué el operador ORDER BY no se evalúa de forma remota.

Consideraciones:

- Opciones de servidor. ¿Cómo afectan los valores de las opciones del servidor COLLATING_SEQUENCE y VARCHAR_NO_TRAILING_BLANKS dónde se evalúa el predicado?
- ¿Se evalúa la entrada de la operación ORDER BY remotamente? Si la respuesta es no, examine la entrada.
- ¿Contiene la cláusula ORDER BY una expresión de caracteres? Si la respuesta es sí, ¿tiene el origen de datos remoto una secuencia de clasificación distinta que la secuencia de clasificación del servidor federado?
- ¿Tiene el origen de datos restricciones sobre este operador? Por ejemplo, ¿existe un número limitado de elementos ORDER BY? ¿Restringe el origen de datos la especificación de columna a la lista ORDER BY?

Por qué no se evalúa completamente una sentencia INSERT con fullselect de forma remota

Puede comprobar varios elementos de la subselección para determinar por qué una sentencia remota INSERT con fullselect no se evalúa completamente de forma remota.

Consideraciones:

- ¿Se ha podido evaluar completamente la subselección en el origen de datos remoto? Si no es así, examine la subselección.
- ¿Contiene la subselección un operador set? Si la respuesta es sí, ¿da este origen de datos soporte a operadores set como entrada de una sentencia INSERT?
- ¿Hace referencia la subselección a la tabla de destino? Si la respuesta es sí, ¿permite este origen de datos esta sintaxis?

Por qué no se evalúa completamente una sentencia remota INSERT con cláusula VALUES de forma remota

Puede comprobar la cláusula VALUES y la expresión para determinar por qué una sentencia remota INSERT con una cláusula VALUES no se evalúa completamente de forma remota.

Consideraciones:

- ¿Se puede evaluar completamente la cláusula VALUES en el origen de datos remoto? En otras palabras, ¿contiene una expresión una función no soportada por el origen de datos remoto?
- ¿Implica la expresión una subconsulta escalar? ¿Se da soporte a esta sintaxis?
- ¿Hace referencia la expresión a la tabla de destino? ¿Se da soporte a esta sintaxis?

Por qué una sentencia UPDATE remota y búsqueda no se evalúa completamente de forma remota

Puede comprobar elementos de la cláusula SET y de la condición de búsqueda para determinar por qué una sentencia UPDATE remota y búsqueda no se evalúa completamente de forma remota.

Consideraciones:

- ¿Se puede evaluar completamente la cláusula SET en el origen de datos remoto? En otras palabras, ¿contiene una expresión de actualización una función no soportada por el origen de datos remoto?
- ¿Implica la cláusula SET una subconsulta escalar? ¿Permite el origen de datos esta sintaxis?
- ¿Se puede evaluar completamente la condición de búsqueda en el origen de datos remoto? Si la respuesta es no, examine en su lugar la condición de búsqueda.
- ¿Hacen referencia la condición de búsqueda o la cláusula SET a la tabla de destino? ¿Permite el origen de datos esta sintaxis?
- ¿Hacen la condición de búsqueda o la cláusula SET referencia a la tabla de destino con correlación? ¿Permite el origen de datos esta sintaxis?

Por qué una sentencia UPDATE posicionada no se evalúa completamente de forma remota

Esto sucede cuando la base de datos federada selecciona evaluar la expresión de actualización de forma local antes de enviar la sentencia UPDATE al origen de datos. Este enfoque no debe afectar de forma significativa el rendimiento.

Consideraciones:

- ¿Se puede evaluar completamente la cláusula SET en el origen de datos remoto? En otras palabras, ¿contiene una expresión de actualización una función no soportada por el origen de datos remoto?
- ¿Implica la cláusula SET una subconsulta escalar? ¿Permite el origen de datos esta sintaxis?

Por qué una sentencia DELETE remota y búsqueda no se evalúa completamente de forma remota

Puede seleccionar elementos de la condición de búsqueda para determinar por qué una sentencia DELETE remota y búsqueda no se evalúa completamente de forma remota.

Consideraciones:

- ¿Se puede evaluar completamente la condición de búsqueda en el origen de datos remoto? Si la respuesta es no, examine en su lugar la condición de búsqueda.
- ¿Hace referencia la condición de búsqueda a la tabla de destino? ¿Permite el origen de datos esta sintaxis?
- ¿Hace referencia la condición de búsqueda a la tabla de destino con correlación? ¿Permite el origen de datos esta sintaxis?

Actualizaciones y personalización de orígenes de datos

Cuando se actualice o personalice orígenes de datos, debe actualizar la información de catálogo global.

El compilador de SQL se basa en información que está almacenada en el catálogo global para proporcionarle las capacidades de SQL de los orígenes de datos. Esta información se debe actualizar periódicamente. Las prestaciones de SQL de los orígenes de datos pueden cambiar en nuevas versiones de los orígenes de datos. Cuando actualice o personalice orígenes de datos, actualice la información de catálogo global de manera que el compilador de SQL utilice la información más actualizada.

Utilice sentencias DDL de SQL, como por ejemplo CREATE FUNCTION MAPPING y ALTER SERVER, para actualizar el catálogo.

Envío de predicados con plantillas de función

En un sistema federado, cada origen de datos remota tiene sus propias funciones. La mayoría de estas funciones tienen funciones de DB2 equivalentes y tienen correlaciones de función asociadas por omisión. Sin embargo, es posible que algunas funciones del origen remoto no tengan funciones equivalentes en el servidor federado. En consecuencia, sólo los datos remotos pueden ejecutar estas funciones. Para grabar consultas que utilicen estas funciones, debe crear una plantilla de función en el servidor federado.

Una plantilla de función actúa como una descripción local de la función remota. Debe crear una plantilla de función con la sentencia CREATE FUNCTION utilizando la cláusula AS TEMPLATE. No existe ningún código ejecutable asociado con la plantilla de función en el servidor federado. Cuando se ha definido la plantilla, puede utilizarla para crear una correlación de función, que correlaciona la plantilla de función con su equivalente remoto. A continuación, es posible hacer referencia a la plantilla de función en las sentencias de SQL que se emiten al servidor federado y para que se evalúe la función en el origen de datos.

El optimizador de consultas toma decisiones basadas en costes a fin de determinar dónde se puede evaluar un predicado. Cuando es posible, el optimizador genera un plan para evaluar un predicado con una plantilla de función en el servidor remoto correspondiente. En algunos casos, puede que no sea posible que el

optimizador genere un plan que evalúe la plantilla de función en el origen de datos. Cuando eso pase, es posible que reciba un error SQL0142N con el siguiente mensaje de error:

La sentencia de SQL no está soportada.

Para evitar este error, se puede sobrescribir la consulta para habilitar el envío mientras que se mantiene la semántica de la consulta original.

Para que se envíe una plantilla de función, se debe definir con las cláusulas DETERMINISTIC y NO EXTERNAL ACTION.

Capítulo 24. Paralelismo con consultas que hacen referencia a apodos

Las consultas que contienen apodos pueden participar en tres tipos de paralelismo dentro de consultas.

Los tres tipos de paralelismo dentro de consultas son los siguientes:

- Paralelismo de consulta dentro de particiones en configuraciones de una única partición y multiprocesador
- Paralelismo de consulta entre particiones en configuraciones de múltiples particiones
- Paralelismo de consulta mixto que consta tanto de paralelismo intrapartición como de paralelismo interparticiones donde cada partición se ejecuta en un sistema SMP

Paralelismo intrapartición con consultas que hacen referencia a apodos

El paralelismo intrapartición hace referencia al proceso de dividir una consulta en varias partes concurrentes que son ejecutadas en paralelo por varios procesos en una única partición de base de datos.

En consultas federadas, la parte de una consulta que implica datos locales se puede ejecutar en paralelo mientras que la parte que implica apodos se ejecuta en serie, utilizando un único proceso de agente.

Cuando varios procesadores pueden funcionar en varias partes de la consulta, el rendimiento de las consultas que hacen referencia a las tablas locales y apodos puede mejorar.

El parámetro de configuración de base de datos DFT_DEGREE y el registro especial CURRENT DEGREE controlan el grado de paralelismo dentro de particiones.

Habilitación del paralelismo intrapartición con consultas que hacen referencia a apodos

Para consultas que hacen referencia a tablas y apodos locales en un entorno de varios procesadores, puede habilitar el paralelismo dentro de particiones. A continuación, el servidor federado puede procesar las tablas locales en paralelo.

Restricciones

El sistema federado puede procesar sólo la parte de una consulta que hace referencia a tablas locales en paralelo. La partición de coordinador procesa todas las operaciones en la partición remota de una consulta en serie.

Procedimiento

Para habilitar el paralelismo dentro de particiones:

1. Establezca el parámetro de configuración de base de datos INTRA_PARALLEL en YES.
2. Establezca el parámetro de configuración de base de datos MAX_QUERYDEGREE en un valor mayor de 1.
3. Establezca el parámetro de configuración de base de datos DFT_DEGREE en un valor mayor de 1 o establezca el registro especial CURRENT DEGREE. Si establece el parámetro DFT_DEGREE en ANY, el nivel por omisión de paralelismo intrapartición equivale al número de procesadores en el sistema.

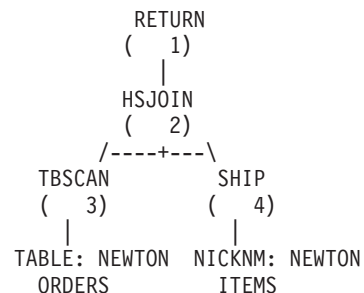
Paralelismo intrapartición con consultas que hacen referencia a apodos - ejemplos de planes de acceso

Puede utilizar el recurso de DB2 Explain para ver el plan de acceso que utiliza el optimizador durante el proceso de consultas. Los ejemplos siguientes muestran cómo el optimizador accede a los datos de apodo en un entorno de paralelismo dentro de particiones.

Ejemplo 1: Sin soporte de paralelismo

En este ejemplo, el servidor federado procesa la unión de la tabla local, ORDERS, y el apodo, ITEMS, en serie. No se utiliza ningún paralelismo dentro de particiones.

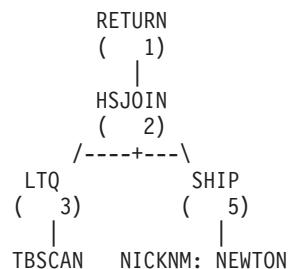
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

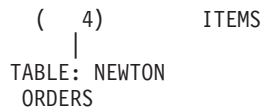


Ejemplo 2: Con soporte de paralelismo

En este ejemplo de una unión, la consulta se puede ejecutar más rápido haciendo que la tabla local lea en paralelo antes de la unión serie con el apodo. El operador LTQ indica dónde se introduce el paralelismo en el plan.

```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```





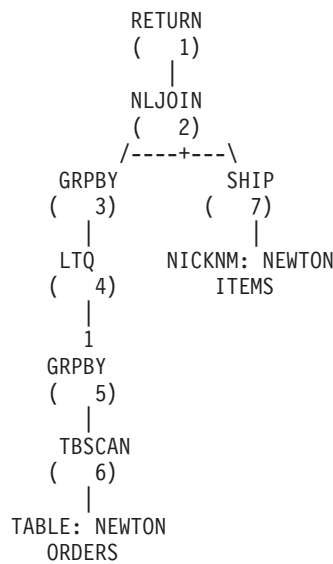
Ejemplo 3: Paralelismo intrapartición con agregación

En este ejemplo, la base de datos agrega los datos de tabla local en paralelo en la partición, mejorando la ejecución de la agregación. La unión de la tabla local y el apodo se produce en serie.

```

SELECT *
FROM ITEMS A
WHERE ID =
(SELECT MAX(ID)
FROM ORDERS
WHERE NUMBER = 10)

```



Paralelismo interparticiones con consultas que hacen referencia a apodos

El paralelismo interparticiones hace referencia al proceso de dividir una única consulta en varias partes que se ejecutan en paralelo en distintas particiones de una base de datos particionada.

En consultas que hacen referencia a datos locales y remotos, el servidor federado puede distribuir los datos remotos a cada una de las particiones locales. La Figura 10 en la página 252 y la Figura 11 en la página 252 muestran el concepto de paralelismo interparticiones que implica orígenes de datos locales y remotos.

La Figura 10 en la página 252 muestra cómo se procesa este tipo de consulta sin paralelismo interparticiones. Los datos de apodo remotos y los datos particionados locales se procesan en serie en la partición de coordinador. Esta técnica no explota la potencia paralela de las particiones de base de datos porque la mayor parte del proceso se realiza en una única partición. Si los volúmenes de datos son muy grandes, es probable que esta técnica dé como resultado consultas de largo tiempo de ejecución.

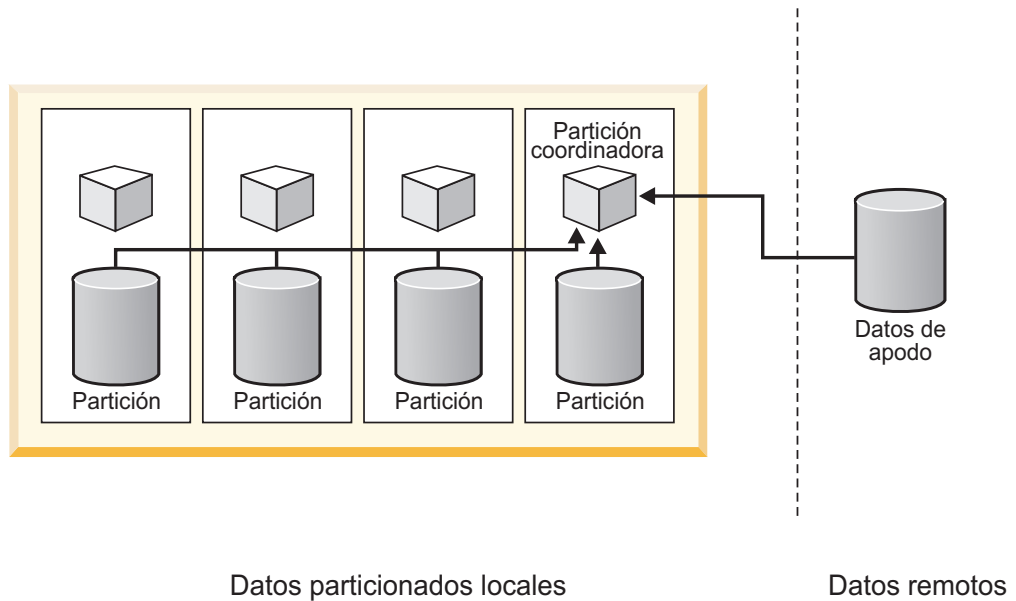


Figura 10. Consulta sin paralelismo interparticiones de orígenes de datos locales y remotos

La Figura 11 muestra cómo se produce el proceso cuando el optimizador distribuye los datos de apodo a las particiones. La partición de coordinador capta los datos de apodo y distribuye los datos a las particiones de base de datos para el proceso en paralelo. Cuando se completa el proceso en paralelo, los resultados se envían de vuelta a la partición de coordinador para el proceso final antes de que se devuelvan a la aplicación.

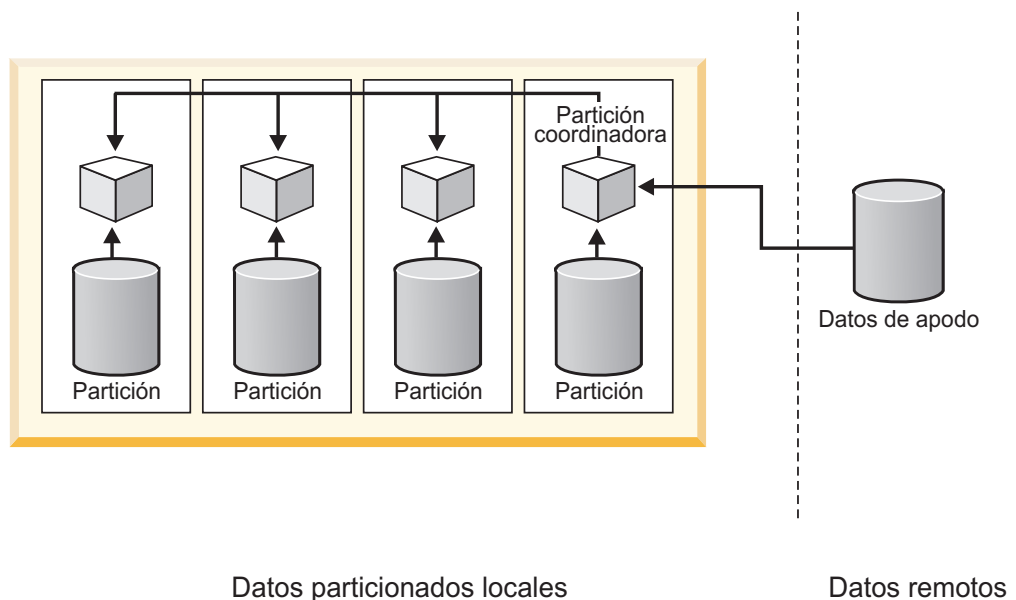


Figura 11. Consulta con paralelismo interparticiones de orígenes de datos locales y remotos

La Figura 12 en la página 253 y la Figura 13 en la página 253 muestran el concepto de paralelismo interparticiones que implica sólo orígenes de datos remotos.

La Figura 12 en la página 253 muestra el proceso en serie de los datos de apodo remotos en la partición de coordinador. La partición de coordinador, que también

actúa como el servidor federado, recupera los datos de apodo y los procesa en serie.

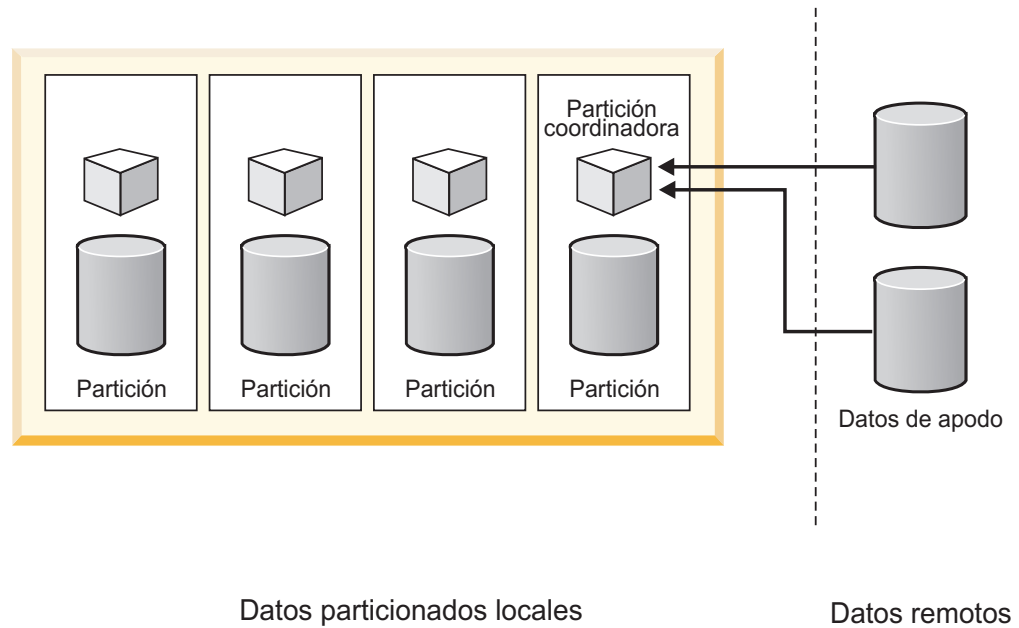


Figura 12. Consulta sin paralelismo interparticiones que hace referencia sólo a orígenes de datos remotos.

La Figura 13 muestra la partición de coordinador que distribuye los datos en todo un grupo de particiones computacionales. Los grupos de particiones computacionales permiten al optimizador generar planes de acceso que distribuyen datos de apodo entre las particiones de un servidor de bases de datos particionado para el proceso en paralelo.

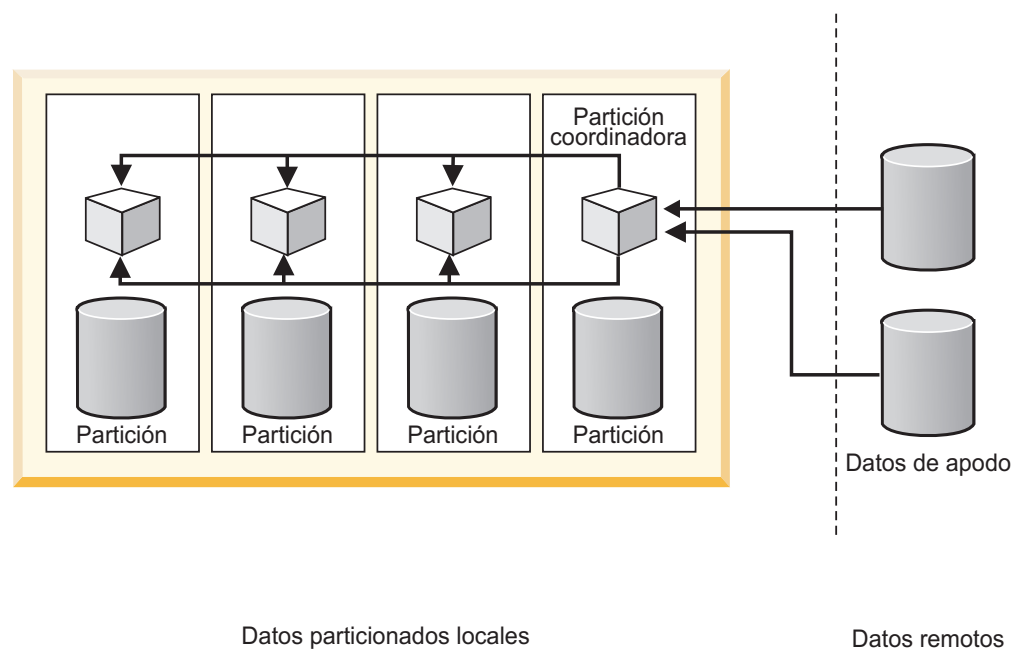


Figura 13. Consulta con paralelismo interparticiones que hace referencia sólo a orígenes de datos remotos.

Independientemente del plan que seleccione el optimizador, el acceso a los datos de apodo siempre se produce en serie desde la partición de coordinador.

Habilitación del paralelismo interparticiones con consultas que hacen referencia a apodos

Puede configurar un servidor federado particionado de forma que las consultas que impliquen apodos se puedan ejecutar potencialmente en paralelo en varias particiones. Es posible que la ejecución en paralelo reduzca significativamente el tiempo transcurrido de consultas federadas en un entorno particionado.

Restricciones

Sólo se pueden ejecutar en paralelo las partes de una consulta que hacen referencia a apodos que utilizan derivadores con limitaciones. Las partes de una consulta que hacen referencia a apodos que utilizan derivadores fiables no se pueden ejecutar en paralelo.

Acerca de esta tarea

En un entorno de base de datos particionado, las consultas federadas se pueden ejecutar en paralelo en las siguientes condiciones:

- Implican una combinación de apodos definidos utilizando un derivador con limitaciones y tablas particionadas locales
- Implican apodos definidos utilizando un derivador con limitaciones y un grupo de particiones computacionales definido.

No necesita establecer ningún parámetro de base de datos ni ningún parámetro de configuración de base de datos en un entorno particionado para hacer que el paralelismo intrapartición esté disponible para consultas federadas.

Procedimiento

Para habilitar el paralelismo entre particiones:

1. Emita la sentencia CREATE WRAPPER o ALTER WRAPPER con la opción DB2_FENCED establecida en Y.
2. **Opcional:** configure un grupo de particiones computacionales, si está interesado en habilitar el paralelismo para partes de consultas que impliquen sólo apodos. Si las consultas implican una combinación de apodos y tablas particionadas locales, no necesita configurar un grupo de particiones computacionales.

Paralelismo interparticiones con consultas que hacen referencia a apodos - ejemplos de planes de acceso

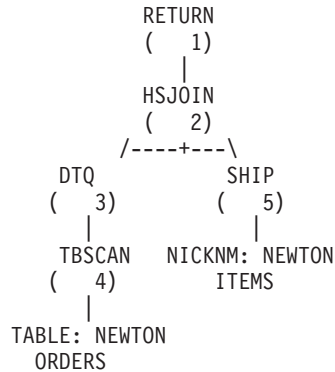
Puede utilizar el recurso de DB2 Explain para ver el plan de acceso que utiliza el optimizador durante el proceso de consultas. Los ejemplos siguientes muestran cómo el optimizador accede a datos de apodo en un entorno de paralelismo entre particiones.

Ejemplo 1: Modalidad fiable

En este ejemplo, el apodo utiliza un derivador fiable. La base de datos realiza en serie la unión entre la tabla local y el apodo en la partición de coordinador. La base de datos lleva los datos locales, que se distribuyen en dos particiones, a la partición de coordinador. A continuación, el servidor federado une los datos locales

con los datos de apodo. La base de datos une en serie apodos definidos utilizando un derivador fiable en la partición de coordinador. La base de datos no puede distribuir los datos entre múltiples particiones para crear una unión en paralelo.

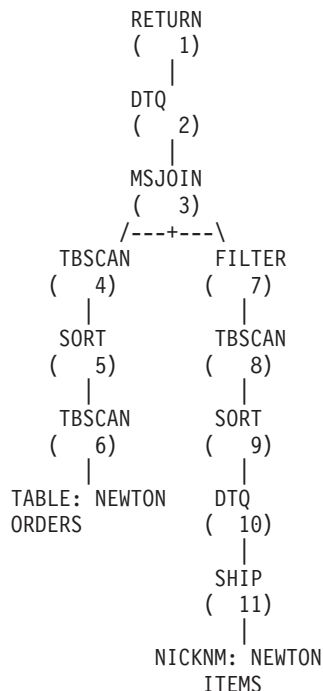
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Ejemplo 2: Modalidad con limitaciones

En este ejemplo, el apodo utiliza un derivador con limitaciones. El servidor federado distribuye los datos de apodo a otras particiones y realiza la unión con los datos locales en paralelo. El operador de DTQ (cola de tabla distribuida) situado sobre SHIP indica que los datos de apodo se distribuyen en las particiones locales utilizando particionamiento hash para conseguir una unión en paralelo co-ubicada. En una unión en paralelo co-ubicada, los datos de apodo se distribuyen a las particiones locales de tal forma que los datos locales y de apodo coincidentes para la unión siempre estarán ubicados en la misma partición.

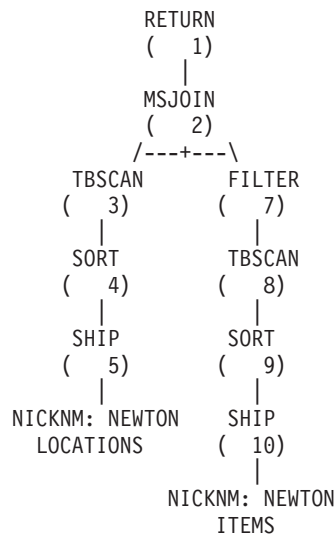
```
SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```



Ejemplo 3: Modalidad con limitaciones sin un grupo de particiones computacionales

En este ejemplo, los dos apodos utilizan un derivador con limitaciones y no hay definido un grupo de particiones computacionales. El servidor federado realiza la unión en la partición de coordinador. El servidor federado no distribuye los datos a otras particiones para su proceso. La falta de operadores de TQ sobre cualquiera de los operadores SHIP indica que los datos de apodo no se distribuyen entre las particiones.

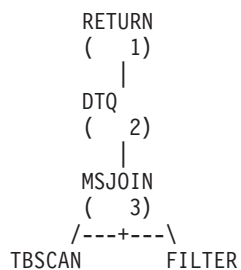
```
SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID1 = B.ID1
```

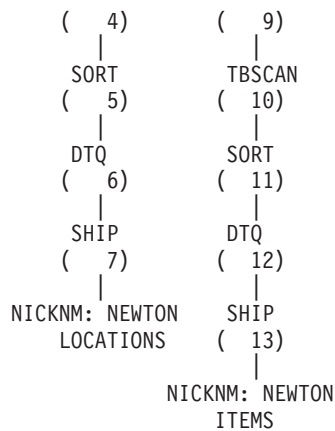


Ejemplo 4: Modalidad con limitaciones con un grupo de particiones computacionales

En este ejemplo, los apodos utilizan derivadores con limitaciones y está definido un grupo de particiones computacionales. En este caso, el optimizador selecciona un plan que distribuye los datos de la partición de coordinador a las otras particiones del grupo de particiones computacionales. Los operadores de DTQ situados sobre ambos apodos realizan una partición hash de los datos remotos entrantes de forma que las claves de unión coincidentes están ubicadas en la misma partición del grupo de particiones computacionales. La unión tiene lugar en paralelo en cada partición y, a continuación, los resultados se recopilan en la partición de coordinador.

```
SELECT *
FROM ITEMS A, LOCATIONS B
WHERE A.ID = B.ID
```





Grupos de particiones computacionales

Un grupo de particiones computacionales define un conjunto de particiones de base de datos que el optimizador puede utilizar para redistribuir datos de apodo de forma dinámica. Un grupo de particiones computacionales es para las partes de las consultas que implican sólo apodos.

La partición de coordinador captura los datos de apodo en serie y, a continuación, redistribuye los datos entre las particiones en el grupo de particiones computacionales, en cuyo punto se produce el proceso en paralelo. La utilización de grupos de particiones computacionales por parte del optimizador a menudo da como resultado mejoras en el rendimiento, especialmente cuando hay grandes cantidades de datos de apodo implicados o las consultas son complejas.

Un grupo de particiones computacionales es un grupo de particiones de base de datos, distinto de `IBMCATNODEGROUP`, que está especificado en el catálogo del sistema, `SYSCAT.DBPARTITIONGROUPS`.

Debe utilizar la variable de registro `DB2_COMPPARTITIONGROUP` para especificar el grupo de particiones computacionales.

Definición de un grupo de particiones computacionales

La definición de un grupo de particiones computacionales permite al optimizador utilizar un plan que distribuye datos de apodo a las particiones del grupo de particiones computacionales. Debe definir un grupo de particiones computacionales para habilitar el paralelismo de consultas dentro de particiones para consultas o partes de consultas que hacen referencia sólo a apodos.

Antes de empezar

Todos los grupos de particiones utilizados para representar el grupo de particiones computacionales en todas las bases de datos de la instancia deben tener el mismo nombre. Puede definir estos grupos de particiones de forma distinta en cada base de datos, pero deben tener el mismo nombre. Por ejemplo, tres bases de datos denominadas `DB1`, `DB` y `DB3` definen un grupo de particiones computacionales que contiene nodos distintos:

- `DB1`: CPG contiene los nodos 1, 2, 3 y 4
- `DB2`: CPG contiene los nodos 49, 50 y 53
- `DB3`: CPG contiene los nodos 78 y 96

Puede establecer la variable db2set en el nombre CPG. El nombre CPG es común a todas las bases de datos, pero el contenido de CPG es distinto para cada base de datos.

Restricciones

El optimizador utiliza grupos de particiones computacionales sólo para las partes de una consulta que hacen referencia a apodos sin hacer referencia a datos locales.

Procedimiento

Para definir un grupo de particiones computacionales, emita el siguiente mandato en la línea de mandatos de DB2.

```
db2set DB2_COMPPARTITIONGROUP=nombre_grupopartición
```

nombre_grupopartición es el nombre del grupo de particiones que desea definir como el grupo de particiones computacionales. El grupo de particiones ya debe estar definido.

El ejemplo siguiente muestra cómo definir el grupo de particiones computacionales, FINANCE3, utilizando la variable de registro DB2_COMPPARTITIONGROUP.

```
db2set DB2_COMPPARTITIONGROUP=FINANCE3
```

Paralelismo interparticiones con consultas que hacen referencia a apodos - expectativas de rendimiento

Para consultas que hagan referencia a una combinación de tablas particionadas locales y apodos, el optimizador puede seleccionar un plan de ejecución que redistribuya datos de apodo entre las particiones adecuadas.

Los planes de redistribución pueden hacer que las consultas se ejecuten más rápidamente si la cantidad de datos de apodo en la unión es más pequeña que la cantidad de datos particionados locales. Si la cantidad de datos de apodo en la unión es considerablemente mayor que los datos locales, es improbable que se utilice un plan paralelo con redistribución de los datos de apodo. Si el optimizador no elige un plan paralelo, el servidor federado realiza las uniones en serie entre apodos y tablas locales en la partición de coordinador.

Para uniones entre dos apodos, un plan de ejecución que distribuye los datos entre todas las particiones de un grupo de particiones computacionales puede ser beneficioso si implica una gran cantidad de datos. La ventaja de procesar la gran unión en paralelo desplaza el coste adicional de redistribuir los datos entre varias particiones. Si la cantidad de datos de apodo es relativamente pequeña, la unión no es lo suficientemente cara para merecer el coste adicional de redistribuir los datos entre particiones. En general, el optimizador selecciona planes de grupos de particiones computacionales si los apodos implicados son grandes; de lo contrario, el servidor federado une los apodos en serie en la partición de coordinador.

Paralelismo mixto con consultas que hacen referencia a apodos

Para consultas que contienen tablas locales o apodos en un entorno particionado, el optimizador puede utilizar tanto el paralelismo interparticiones como el paralelismo dentro de particiones. El paralelismo interparticiones es una opción para el optimizador en un entorno particionado. El paralelismo intrapartición es

una opción, si se ha habilitado en la configuración de la base de datos o en la configuración del gestor de bases de datos.

Para el paralelismo entre particiones, el servidor federado puede distribuir datos remotos entre particiones y procesar datos en paralelo dentro de cada partición.

Para el paralelismo dentro de particiones, se utilizan varios procesos de subagente dentro de una partición para procesar datos locales en paralelo.

Habilitación del paralelismo mixto con consultas que hacen referencia a apodos

Puede mejorar el rendimiento de las consultas que hacen referencia a datos locales y remotos mediante la utilización de paralelismo intrapartición y paralelismo entre particiones.

Procedimiento

Para habilitar el paralelismo interparticiones en un servidor federado particionado:

1. Emita la sentencia CREATE WRAPPER o ALTER WRAPPER con la opción DB2_FENCED establecida en Y.
2. **Opcional:** configure un grupo de particiones computacional para habilitar el paralelismo en uniones de sólo apodos.

Para habilitar el paralelismo interparticiones en un servidor federado:

1. Establezca el parámetro de configuración de base de datos MAX_QUERYDEGREE en un valor mayor de 1.
2. Establezca el parámetro de configuración de base de datos DFT_DEGREE en un valor mayor de 1 o debe establecer el registro especial CURRENT DEGREE. Si establece el parámetro DFT_DEGREE en ANY, el nivel por omisión del paralelismo interparticiones equivale al número de procesadores SMP en el sistema.

Paralelismo mixto con consultas que hacen referencia a apodos - ejemplos de planes de acceso

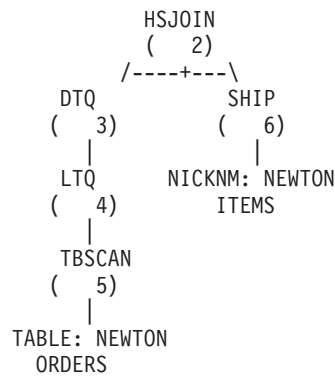
Puede utilizar el recurso de DB2 Explain para ver el plan de acceso que utiliza el optimizador durante el proceso de consultas. Los ejemplos siguientes muestran cómo el optimizador accede a los datos de apodo en un entorno que utiliza tanto el paralelismo intrapartición como el paralelismo entre particiones.

Ejemplo 1: Modalidad fiable

El ejemplo siguiente muestra una unión entre una tabla local y un apodo en modalidad fiable. El servidor federado procesa los datos locales en paralelo en cada partición antes de unir los datos locales con los datos de apodo en la partición de coordinador. El servidor federado no procesa los datos de apodo en paralelo en las particiones o los procesadores en una partición determinada.

```
SELECT *  
FROM ORDERS A, ITEMS B  
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
```

```
RETURN  
( 1)  
|
```

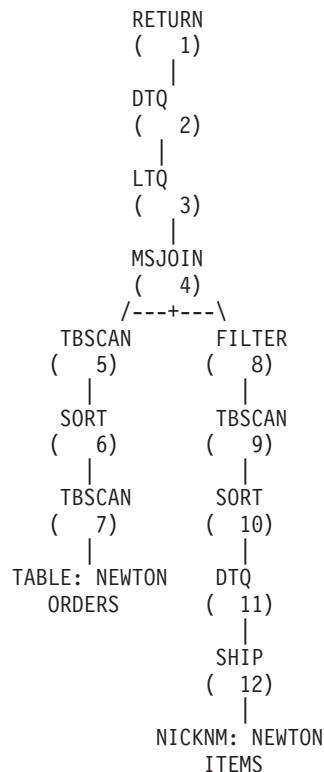


Ejemplo 2: Modalidad con limitaciones

El ejemplo siguiente muestra una unión entre una tabla particionada local y un apodo en modalidad con limitaciones. El servidor federado capta en serie los datos de apodo en la partición de coordinador y, a continuación, distribuye estos datos a las otras particiones de la base de datos. A continuación, los datos se unen en paralelo con los datos locales en todas las particiones de base de datos adecuadas. En cada partición, varios subagentes están leyendo la tabla local y uniéndose a los datos de apodo. Esta operación es paralelismo dentro de particiones, identificado en el plan mediante el operador de LTQ. El resultado se devuelve a la partición de coordinador para el proceso final y se devuelve a la aplicación.

```

SELECT *
FROM ORDERS A, ITEMS B
WHERE A.ID1 = B.ID1 AND B.ITEM = 3
  
```



Capítulo 25. Proceso asíncrono de consultas federadas

La *asincronía* es un método de mejorar el rendimiento de consultas mediante la ejecución de diversas partes de un plan de acceso simultáneamente para reducir el tiempo transcurrido para una consulta determinada.

En un sistema federado, los datos se distribuyen entre los sistemas en varios orígenes de datos y cada sistema tiene sus propios recursos. La asincronía superpone las operaciones que utilizan dichos recursos de manera que múltiples sistemas permanezcan activos a la vez. La superposición de operaciones permite que varias partes de un plan de acceso se ejecuten simultáneamente en lugar de en serie.

Las consultas complejas que implican operaciones que requieren mucho tiempo en orígenes de datos remotos se pueden beneficiar de la asincronía. La asincronía permite que los siguientes tipos de operaciones se lleven a cabo simultáneamente:

- Dos o más operaciones en orígenes de datos remotos
- Operaciones en el servidor federado y como mínimo un origen de datos remoto

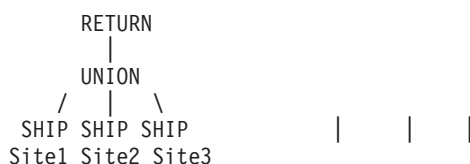
Puesto que las operaciones de consultas consumen recursos, la asincronía puede beneficiar a los sistemas en los que los recursos están inactivos o bien cuando una de los orígenes de datos, o el sistema federado, está trabajando en cualquier punto en el tiempo.

Proceso asíncrono de consultas federadas - ejemplos

Los ejemplos de consultas con una unión y con una unión de fusión muestran la diferencia entre las operaciones de consulta con y sin proceso asíncrono.

Ejemplo: Consulta con una operación de unión

Una consulta simple realiza una operación de unión sobre los datos de tres orígenes de datos distintos. El cálculo necesario para generar los datos en cada origen de datos requiere mucho tiempo. El plan de acceso tiene el aspecto siguiente:



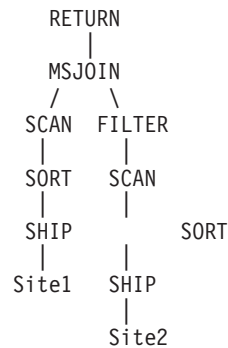
Sin asincronía, la operación de unión lee los datos de las ramas de la consulta leyendo una rama cada vez, de izquierda a derecha. Cuando se leen los datos del servidor Site1, el servidor Site2 y el servidor Site3 están inactivos. Para este ejemplo, cada rama de la unión tarda aproximadamente dos horas en devolver las filas de resultados de cada uno de los sitios. El tiempo total de ejecución de las tres ramas es aproximadamente la suma del tiempo que se tarda en procesar cada rama; en este ejemplo, aproximadamente seis horas.

Con la asincronía, cada rama de la unión se empieza a procesar al mismo tiempo, y los tres servidores remotos están activos de forma simultánea. El tiempo de ejecución de la consulta es aproximadamente equivalente al tiempo de ejecución de

la rama más lenta de la unión. En este ejemplo, el tiempo de ejecución se reduce a aproximadamente dos horas (aproximadamente un 66% más rápido) comparado con las seis horas sin asincronía.

Ejemplo: Consulta con una operación de unión de fusión

Una consulta que une datos de dos orígenes de datos distintos utiliza una operación de unión de fusión (MSJOIN). El plan de acceso del optimizador tiene el aspecto siguiente:



Sin asincronía, el operador de unión de fusión en primer lugar procesa la rama externa (izquierda) y no procesa la rama interna (derecha) hasta que la rama izquierda empieza a devolver filas. Para este ejemplo, cada rama ejecuta una consulta compleja y, por lo tanto, tarda mucho en ejecutarse. El tiempo total aproximado para ejecutar la unión de fusión es la suma del tiempo que se tarda en ejecutar cada rama.

Con asincronía, ambas ramas de la unión de fusión se inician al mismo tiempo, reduciendo en consecuencia el tiempo global de ejecución de la consulta.

Optimización de asincronía

El optimizador de consultas toma decisiones sobre el proceso asíncrono de operaciones remotas en un plan de ejecución de consultas. *Optimización de asincronía* es el proceso mediante el cual el optimizador analiza un plan de ejecución de consulta existente y busca oportunidades para permitir que las operaciones remotas se ejecuten de forma concurrente.

Planes de acceso sin asincronía

En un plan de ejecución, el operador SHIP o RPD define una parte del plan ejecutado en un origen de datos remoto.

Sin asincronía, el operador SHIP o RPD se activa e inicializa el proceso remoto sólo si otros operadores ubicados sobre el operador SHIP o RPD en el plan de ejecución requieren sus datos.

Planes de acceso optimizados para asincronía

El optimizador puede hacer que se ejecuten de forma asíncrona las operaciones remotas que define el operador SHIP o RPD.

En una operación asíncrona, un operador de cola de tabla (TQ) se inserta directamente sobre el operador SHIP o RPD en el plan de ejecución. El operador de TQ define una parte del plan, denominado *subplan*. Un proceso o hebra

distintos, con su propia memoria, ejecutan el subplan. Un subplan se inicializa de forma inmediata cuando se inicia la consulta.

Puede pensar en el operador de TQ como un conducto entre el operador SHIP o RPD (productor de datos) y el operador situado sobre el mismo (el consumidor de datos) en el plan. Este conducto desempareja la ejecución de SHIP en el subplan situado debajo del mismo desde el plan principal y permite el intercambio asíncrono de los datos entre las dos secciones del plan.

Un operador de TQ que aparece directamente sobre un operador SHIP o RPD en el plan permite que las operaciones remotas que define el operador SHIP o RPD se inicialicen al comienzo de la consulta y que entreguen los resultados al servidor federado de forma asíncrona. Cuando la asincronía es beneficiosa para una operación remota determinada, el optimizador coloca un operador de TQ directamente sobre el operador SHIP o RPD correspondiente en el plan.

Los operadores de TQ aparecen en planes de ejecución para propósitos distintos. Un operador de TQ normalmente significa operaciones paralelas en bases de datos particionadas o en bases de datos habilitadas para el paralelismo entre particiones. Otro tipo de operador de TQ, que permite la ejecución asíncrona de un subplan, se denomina TQ de asincronía (ATQ).

El optimizador hace que un operador SHIP o RPD determinado sea asíncrono cuando:

- El rendimiento de consulta mejorará
- El número de ATQ está por debajo de los límites por servidor y por consulta
- El operador no es aún asíncrono debido a la utilización de otra técnica de optimización
- No se violan las restricciones sobre asincronía.
- La semántica de la consulta no cambia

Planes de acceso - Ejemplos

Los ejemplos de planes de acceso muestran la diferencia entre la ejecución del plan con y sin optimización de asincronía.

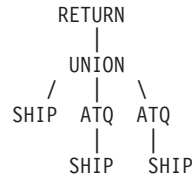
Los primeros dos ejemplos muestran el aspecto de los planes de unión y de unión de fusión en el “Proceso asíncrono de consultas federadas - ejemplos” en la página 261 cuando la asincronía está habilitada.

Por razones de simplificación, los ejemplos muestran planes sólo con operadores SHIP. La optimización de asincronía transforma el plan de la misma forma para operadores RPD que para operadores SHIP. Los operadores SHIP y RPD son intercambiables, a menos que se indique lo contrario.

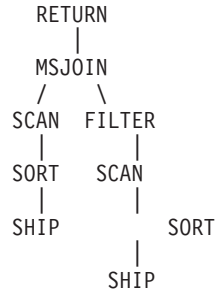
Ejemplo 1a: Plan sin asincronía



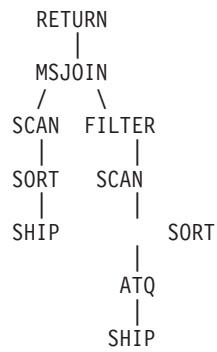
Ejemplo 1b: Plan con asincronía



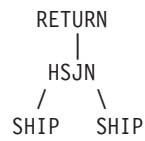
Ejemplo 2a: Plan sin asincronía



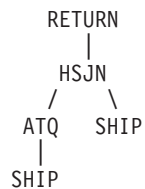
Ejemplo 2b: Plan con asincronía



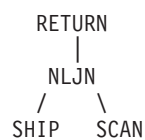
Ejemplo 3a: Plan sin asincronía



Ejemplo 3b: Plan sin asincronía

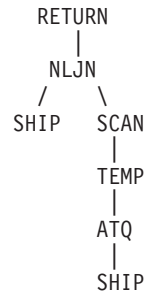


Ejemplo 4a: Plan sin asincronía

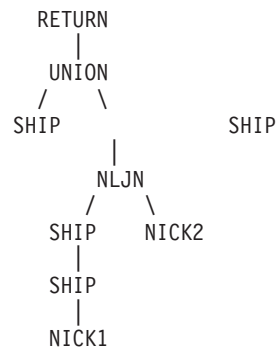




Ejemplo 4b: Plan con asincronía

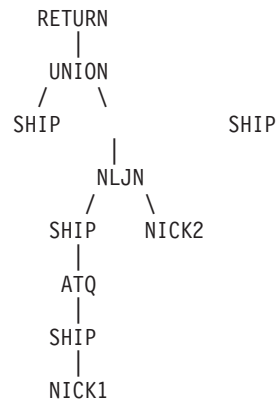


Ejemplo 5a: Plan sin asincronía

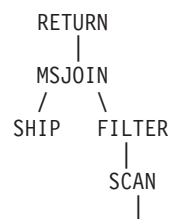


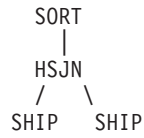
Los RPD no pueden sustituir al par SHIP-SHIP en este plan.

Ejemplo 5b: Plan con asincronía

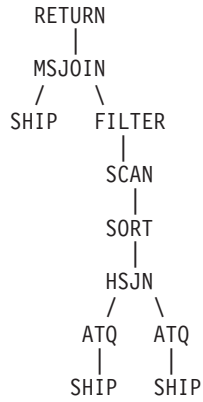


Ejemplo 6a: Plan sin asincronía

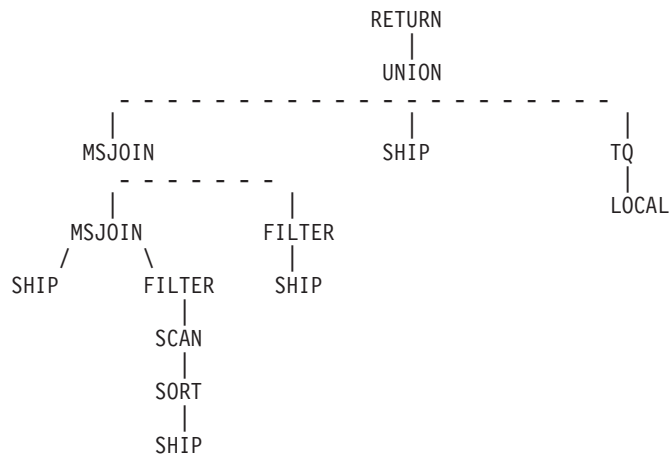




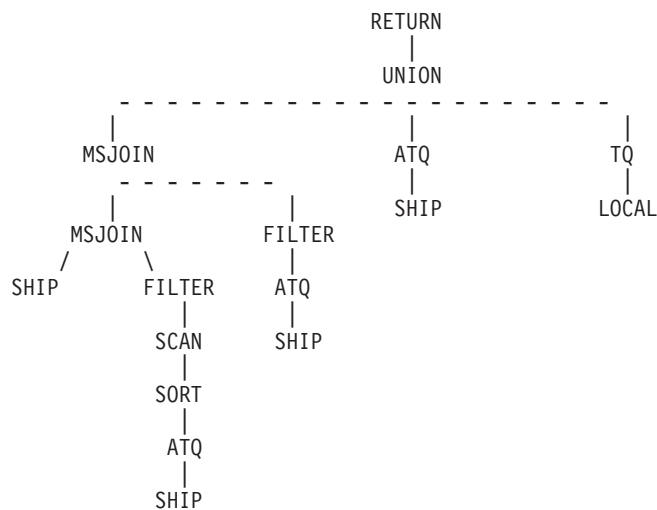
Ejemplo 6b: Plan con asincronía



Ejemplo 7a: Plan sin asincronía



Ejemplo 7b: Plan con asincronía



Control del consumo de recursos

Además de habilitar la ejecución asíncrona de una consulta remota, el operador de ATQ afecta al servidor federado y a los orígenes de datos remotos.

Debido a que cada operador de ATQ crea un nuevo proceso y consume más memoria (para la colocación en el almacenamiento intermedio), es posible que la inserción de numerosos ATQ en un plan de ejecución utilice demasiados recursos en el servidor federado. Además, si varios operadores SHIP o RPD en una consulta se ejecutan en un origen de datos remoto determinado, hacer varios de los operadores asíncronos abre varios cursores concurrentes en ese origen de datos y puede producir una carga inaceptable.

Para controlar el consumo de recursos en el sistema federado, o en los orígenes de datos, puede establecer parámetros de configuración. Los parámetros definen límites en el número total de ATQ permitidas dentro de una consulta o en el número de ATQ permitidas en cada servidor dentro de una consulta.

Habilitación de la optimización de asincronía

Para habilitar la optimización de asincronía, debe especificar el número de operadores de TQ de asincronía para una consulta determinada y establecer una opción del servidor para el origen de datos.

Restricciones

La optimización de asincronía requiere lo siguiente:

- Un sistema federado con la característica de particionamiento de bases de datos (DPF) que incluya más de una partición lógica de base de datos.
- Acceso a orígenes de datos a través de derivadores con limitaciones.

Esta optimización no da soporte a estos objetos:

- Apodos en un origen de datos a los que se accede mediante un derivador fiable.
- Consultas con operaciones de inserción, actualización o supresión.

Procedimiento

Para habilitar el proceso asíncrono:

1. Establezca uno o varios de los parámetros siguientes:
 - Establezca el parámetro de configuración del gestor de bases de datos `FEDERATED_ASYNC` en un valor entre 0 y 32767 inclusive o en `ANY`. `MAXAGENTS` es un parámetro de configuración de base de datos que especifica el número máximo de ATQ permitidas por consulta. El valor `ANY` permite que el optimizador determine el número de ATQ para un plan de acceso determinado. El valor por omisión es 0.
 - De forma opcional, establezca las opciones de precompilación y de vinculación de `FEDERATED_ASYNC` para las sentencias estáticas en el paquete para alterar temporalmente el valor del parámetro de configuración para la consulta. El valor por omisión es 0.
 - Opcionalmente, establezca el registro especial `CURRENT FEDERATED ASYNCHRONY` para alterar temporalmente de forma dinámica el valor del parámetro de configuración del gestor de bases de datos y la opción de vinculación para la consulta.

Estos parámetros forman la siguiente jerarquía:

- a. Registro especial
- b. Opción de vinculación
- c. Parámetro de configuración del gestor de bases de datos

El valor del registro especial, si se especifica, tiene prioridad sobre la opción de vinculación, que a su vez tiene prioridad sobre el parámetro de configuración del gestor de bases de datos.

2. Establezca la opción del servidor `DB2_MAX_ASYNC_REQUESTS_PER_QUERY` en un valor numérico. Esta opción del servidor especifica el número máximo de peticiones asíncronas que permite el servidor para la consulta.

El rango para la opción del servidor es de -1 a 64000.

- El valor por omisión es 1. Por lo tanto, un operador SHIP o un operador RPD que pertenece a un servidor se considera para la asincronía en una consulta.
- El valor por omisión para el origen de datos ODBC sólo es 0. La asincronía requiere varios cursores por conexión. Este valor debe ser 0 para el derivador ODBC a menos que el origen de datos dé soporte a varios cursores por conexión.

Consideraciones sobre los ajustes para la optimización de asincronía

Cuando la asincronía está habilitada, necesita considerar varios factores que afectan al rendimiento.

Si el sistema tiene proceso disponible, memoria y recursos de CPU, la habilitación de la asincronía puede mejorar el rendimiento de las consultas federadas. La habilitación de la asincronía también puede aumentar la utilización de sistemas de orígenes remotos por parte del servidor federado, porque un origen remoto puede potencialmente procesar más de una petición a la vez en nombre de una consulta federada. Si el sistema tiene restricciones de recursos, la asincronía puede degradar el rendimiento.

Puede ajustar el sistema para la asincronía cambiando los parámetros de configuración para conseguir el grado de asincronía que se adapte mejor al sistema.

Cada TQ asíncrona que la optimización de asincronía introduce en un plan requiere un subagente adicional. Si hay suficientes subagentes disponibles en el sistema, considere ajustar el parámetro de configuración del gestor de bases de datos `MAXAGENTS`.

Restricciones sobre la optimización de asincronía

Cuando el optimizador aplica optimización de asincronía en una consulta determinada, se aplican algunas restricciones al número de ATQ que se pueden utilizar en el plan de ejecución de consultas.

El número de SHIP o RPD aptos para ser emparejados con una ATQ en un plan y que se benefician de asincronía puede ser mayor que el máximo que ha establecido el parámetro `FEDERATED_ASYNC` o el límite por servidor de la opción del servidor `DB2_MAX_ASYNC_REQUESTS_PER_QUERY`. En este caso, el optimizador selecciona SHIP o RPD para emparejarlos con una ATQ de tal forma que:

- El número total de ATQ en el plan es inferior que o igual al valor establecido para los parámetros siguientes, en el orden listado:

1. Registro especial FEDERATED ASYNCHRONY, si se especifica
 2. Opción de vinculación o precompilación FEDERATED_ASYNC, si se ha especificado
 3. Parámetro FEDERATED_ASYNC
- El número total de ATQ para un servidor determinado es inferior que o igual a la opción del servidor DB2_MAX_ASYNC_REQUESTS_PER_QUERY para ese servidor.
 - Los SHIP o RPD aptos se pueden beneficiar de ser emparejados con una ATQ y mejorar el rendimiento de consulta.

Determinación de si la optimización de asincronía se aplica a una consulta

Para determinar si la optimización de asincronía se aplica a una consulta determinada, puede utilizar uno de los métodos siguientes para comprobar el plan de acceso para la consulta y comprobar si existe un operador específico.

Puede comprobar cualquiera de las siguientes salidas para la consulta en cuestión:

- Salida de db2exfmt
- Salida de Visual Explain
- Salida de dynexpln

Cada salida muestra peticiones de consultas asíncronas en el plan de acceso como ATQ. La propiedad 'origen' de la descripción de la ATQ muestra 'Asynchrony'.

El siguiente fragmento de plan muestra cómo se utiliza el operador de ATQ y muestra sus propiedades detalladas.

```

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

```

- show the origin of the ATQ as ASYNCHRONY:

```

4) TQ      : (Table Queue)
Cumulative Total Cost: 511.795
Cumulative CPU Cost: 2.79486e+06

```

Cumulative I/O Cost: 16
Cumulative Re-Total Cost: 68.9489
Cumulative Re-CPU Cost: 1.72372e+06
Cumulative Re-I/O Cost: 0
Cumulative First Row Cost: 30.8308
Cumulative Comm Cost: 18.2176
Cumulative First Comm Cost: 0
Estimated Bufferpool Buffers: 16
Remote communication cost: 538.297

Arguments:

JN INPUT: (Join input leg)
OUTER
LISTENER: (Listener Table Queue type)
FALSE
TQMERGE : (Merging Table Queue flag)
FALSE
TQORIGIN: (Table Queue Origin type)
ASYNCHRONY
TQREAD : (Table Queue Read type)
READ AHEAD
TQSEND : (Table Queue Write type)
DIRECTED
UNIQUE : (Uniqueness required flag)
FALSE

Si no encuentra ningún operador de ATQ en el plan de acceso para una consulta, verifique si se satisfacen las siguientes condiciones:

- El sistema es un sistema federado que está habilitado para DPF.
- La consulta accede a apodos en un origen de datos al que se accede mediante un derivador con limitaciones.
- Se habilita la asincronía utilizando el parámetro de configuración del gestor de bases de datos `FEDERATED_ASYNC`, el registro especial `CURRENT FEDERATED ASYNCHRONY` o la opción de vinculación `FEDERATED_ASYNCHRONY`.
- La opción del servidor `DB2_MAX_ASYNC_REQUESTS_PER_CONNECTION` se establece en un valor distinto de cero para los apodos en cada servidor.

Capítulo 26. Optimización global

El compilador de SQL funciona en tres fases, que ayudan a producir una estrategia de acceso óptima para evaluar una consulta que hace referencia a un origen de datos remoto. Estas fases son análisis de envío, optimización global y generación de SQL remoto.

Para una consulta enviada a la base de datos federada, es posible que la estrategia de acceso implique dividir la consulta original en una serie de fragmentos de consulta y, a continuación, combinar los resultados de estos fragmentos de consulta.

Utilizando la salida de la fase de análisis de envío como una recomendación, el optimizador de consulta decide dónde se evalúa cada operación. Una operación se puede evaluar localmente en el servidor federado o de forma remota en el origen de datos. La decisión se basa en la salida del modelo de coste sofisticado que utiliza el optimizador. Este modelo determina:

- El coste para evaluar la operación
- El coste para transmitir los datos o mensajes entre el servidor federado y el origen de datos

El objetivo de la optimización global es producir un plan de acceso que optimiza las operaciones de consulta en todos los orígenes de datos globalmente, en todo el sistema federado. Un plan de acceso que es óptimo globalmente tiene como mínimo un coste global de ejecución en un sistema federado. La fase de generación de SQL remoto convierte a la inversa el plan óptimo globalmente en fragmentos de consulta que se ejecutan como orígenes de datos individuales.

El compilador de SQL tiene una base de conocimientos que contiene características de orígenes de datos soportados y metadatos sobre los datos en estos orígenes de datos. El optimizador no genera SQL, fragmentos de consulta o sugerencias de plan que el origen de datos remoto no pueda entender o aceptar.

Muchos factores pueden afectar la salida de optimización global y, por lo tanto, afectar el rendimiento de la consulta. Los factores clave son las características del servidor y las características del apodo.

Los derivadores relacionales y no relacionales difieren en los detalles acerca de cómo se produce un plan de acceso, pero el concepto y el efecto final son los mismos.

Características de servidor que afectan a la optimización global

Cuando se crea o altera una definición de servidor, algunas de las opciones que elija pueden afectar al rendimiento de la consulta.

Al optimizador de consultas se le proporciona información sobre las características de servidor de orígenes de datos mediante valores de opciones de servidor. Los valores de opciones de servidor forman parte de la definición de servidor de orígenes de datos. Se pueden establecer opciones de servidor en la sentencia CREATE SERVER, al establecer inicialmente la definición de servidor. Utilice la

sentencia ALTER SERVER para añadir opciones de servidor a una definición de servidor existente. Los valores de opciones de servidor se almacenan en el catálogo global de bases de datos federadas.

Estas opciones están clasificadas como opciones de ubicación (como por ejemplo el nombre del sistema de origen de datos), opciones de seguridad (como por ejemplo información de autenticación) y opciones de rendimiento (como por ejemplo la proporción de CPU).

Las opciones de rendimiento ayudan al optimizador a determinar si la evaluación de una operación se puede realizar en un origen de datos y si la evaluación de una operación en el origen de datos hace que la ejecución sea más rápida. Las opciones de servidor que afectan al rendimiento que podrían requerir un ajuste son:

- CPU_RATIO
- IO_RATIO
- COMM_RATE
- COLLATING_SEQUENCE
- PLAN_HINTS
- VARCHAR_NO_TRAILING_BLANKS
- NO_EMPTY_STRING

Tenga cuidado al ajustar las opciones de servidor CPU_RATIO, IO_RATIO o COMM_RATE ya que se podrían obtener errores inesperados si el cálculo del coste para una consulta produce desbordamientos o desbordamientos por debajo del nivel. En la mayoría de casos, los valores por omisión para estas opciones son suficientes. Generalmente, asegurarse de que las estadísticas sobre los objetos referenciados en las consultas sean correctas es más importante que ajustar los valores de estas opciones de servidor.

Proporción relativa de velocidad de CPU

Este valor indica la proporción entre la velocidad de CPU del servidor federado y la velocidad de CPU del servidor en el que está ubicado el origen de datos remoto.

Este valor se define como la velocidad de CPU del servidor federado dividida por la velocidad de CPU del servidor para el origen de datos remoto. Por ejemplo, si la velocidad de CPU para el servidor federado es el doble que la velocidad de CPU para el servidor remoto, CPU_RATIO se debe establecer en 2. Si la velocidad de CPU para el servidor federado es sólo un tercio de la velocidad de CPU para el servidor remoto, CPU_RATIO se debe establecer en 0.33.

Cuando no establece explícitamente la opción del servidor de proporción de CPU, el optimizador federado utiliza un valor por omisión de 1, que indica que la velocidad de CPU federada y la velocidad de CPU de origen de datos son iguales.

Una proporción baja indica que la CPU del servidor de orígenes de datos es más rápida que la CPU del servidor federado. Para proporciones bajas, el optimizador considerará enviar las operaciones con un uso intensivo de CPU al origen de datos. Una proporción baja es un valor que es inferior a 1.

Proporción relativa de velocidad de E/S

Este valor indica la proporción entre la velocidad de E/S del servidor federado y la velocidad de E/S del servidor en el que está ubicado el origen de datos remoto.

Este valor se define como la velocidad de E/S para el servidor federado dividida por la velocidad de E/S para el servidor remoto. Por ejemplo, si la velocidad de E/S para el servidor federado es el doble que la velocidad de E/S para el servidor remoto, IO_RATIO se debe establecer en 2. Pero si la velocidad de I/O para el servidor federado es la mitad que el servidor remoto, IO_RATIO se debe establecer en 0.5.

Cuando no establece explícitamente la opción del servidor de proporción de E/S, el optimizador federado utiliza un valor por omisión de 1, que indica que las velocidades de E/S de tanto el servidor federado como del servidor remoto son iguales.

Una IO_RATIO baja de menos de uno indica que el servidor remoto tiene una velocidad de E/S más alta que el servidor federado. En este caso, el optimizador tenderá a favorecer el envío de operaciones intensivas de E/S al origen de datos remoto. Una proporción baja es un valor que es inferior a 1.

Velocidad de comunicaciones entre el servidor federado y el origen de datos

Una velocidad de comunicaciones baja indica comunicación lenta de red entre el servidor federado y el origen de datos.

El valor de la opción del servidor COMM_RATE determina la velocidad de comunicaciones. COMM_RATE representa la velocidad de la conexión de red entre el servidor de orígenes de datos y el servidor federado. La velocidad se mide en megabytes por segundo. El valor por omisión es 2 MBPS.

Las velocidades bajas de comunicación hacen que el optimizador de consultas reduzca el número de mensajes enviados a o desde este origen de datos. Si la opción del servidor COMM_RATE está establecida en un número muy bajo, el optimizador produce una consulta que requiere tráfico de red mínimo.

Secuencia de clasificación de origen de datos

La secuencia de clasificación que elija puede afectar al rendimiento de la base de datos federada. Puede utilizar la opción del servidor COLLATING_SEQUENCE para indicar si una secuencia de clasificación de origen de datos coincide con la secuencia de clasificación de la base de datos federada local.

El servidor federado puede enviar un proceso dependiente del orden que implique datos de caracteres al origen de datos, si la opción del servidor COLLATING_SEQUENCE indica que la secuencia de clasificación del origen de datos y la base de datos federada coinciden. Si una secuencia de clasificación del origen de datos no coincide con la secuencia de clasificación de base de datos federada, el optimizador considera que los datos que se recuperan de este origen de datos están desordenados. La base de datos federada recuperará los datos relevantes y realizará todo el proceso dependiente del orden sobre los datos de caracteres de forma local, lo que ralentizará la consulta y afectará al rendimiento.

Sugerencias de plan remoto

Las sugerencias de plan son fragmentos de sentencia que proporcionan información adicional a los optimizadores de orígenes de datos.

Utilice la opción del servidor PLAN_HINTS para generar sugerencias de plan remoto. Esta información puede, para algunos tipos de consulta, mejorar el

rendimiento de consulta. Las sugerencias de plan pueden ayudar al optimizador de orígenes de datos a decidir si utilizar un índice, qué índice utilizar o qué secuencia de unión de tablas utilizar.

Debe ejecutar algunas pruebas para determinar si esta opción del servidor mejorará el rendimiento de las consultas.

No puede codificar sus propias sugerencias de plan en una consulta.

Si estas sugerencias de plan están habilitadas, la consulta que se ha enviado al origen de datos contiene información adicional. Por ejemplo, una sentencia enviada a un optimizador de Oracle con sugerencias de plan podría tener el aspecto siguiente:

```
SELECT /*+ INDEX (table1, t1index)*/  
  col1  
FROM table1
```

La sugerencia de plan es la serie `/*+ INDEX (table1, t1index)*/`

Características de apodo que afectan a la optimización global

Existen varios factores específicos del apodo que pueden afectar a la optimización global, incluyendo la información de índice y las estadísticas de catálogo global.

Es importante que la información de índice y los datos estadísticos de catálogo global disponibles al compilador de SQL sean actuales.

Especificaciones de índice

El compilador de SQL utiliza información de índice para optimizar las consultas.

La información de índice para una tabla de origen de datos sólo se adquiere cuando se crea el apodo para esa tabla. Una vez que se ha creado el apodo, los cambios realizados al índice en esa tabla de origen de datos no se actualizan en el servidor federado. Cuando cambia la información del índice remoto, puede actualizar la información de índice almacenada en el servidor federado descartando el apodo para la tabla y creando de nuevo el apodo. De forma alternativa, si se añade un nuevo índice para la tabla de origen de datos, puede definir una especificación de índice para el apodo en el servidor federado.

La información de índice no se recopila para apodos en objetos que no tienen índices tales como vistas, sinónimos u objetos de orígenes de datos no relacionales.

Si un objeto que tiene un apodo definido para el mismo no tiene un índice, puede crear una especificación de índice para el mismo. Las especificaciones de índice crean una definición de índice en el catálogo global. La especificación de índice no es un índice real. Utilice la sentencia `CREATE INDEX` con la cláusula `SPECIFICATION` para crear una especificación de índice. La sintaxis para crear una especificación de índice en un apodo es similar a la sintaxis para crear un índice en una tabla local.

Tenga en cuenta la creación de especificaciones de índice cuando:

- Una tabla adquiere un nuevo índice.
- Cree un apodo para un objeto de origen de datos que no contenga índices tal como una vista o un sinónimo.

Cuando crea una especificación de índice (SPECIFICATION ONLY) en un apodo y especifica que el índice es exclusivo, la base de datos federada no verifica que los valores de columna en la tabla remota sean exclusivos. Si los valores de columna remotos no son exclusivos, es posible que las consultas contra el apodo que incluyan la columna de índice den como resultado datos incorrectos o que den como resultado errores.

Tenga en cuenta sus necesidades antes de emitir sentencias CREATE INDEX...SPECIFICATION ONLY en un apodo para una vista de origen de datos:

- Si la vista remota es una sentencia SELECT simple en una tabla de origen de datos con un índice, la creación de una especificación de índice en el apodo que coincida con el índice en la tabla de origen de datos puede mejorar significativamente el rendimiento de la consulta.
- Si se crea una especificación de índice para una vista remota que no es una sentencia SELECT simple (por ejemplo, una vista creada uniendo dos tablas), es posible que el rendimiento de la consulta se vea afectado.

Por ejemplo, tenga en cuenta una especificación de índice que se cree para una vista remota que sea una unión de dos tablas. El optimizador puede seleccionar esa vista como el elemento interno en una unión de bucle anidado. Es posible que el rendimiento de la consulta sea bajo porque la unión se evaluará varias veces. Una alternativa consiste en crear apodos para cada una de las tablas a las que se hace referencia en la vista de origen de datos y crear una vista federada que haga referencia a ambos apodos.

Estadísticas de catálogo global

El catálogo global en el servidor federado contiene información de estadísticas que se utilizan para optimizar las consultas.

El servidor federado confía en las estadísticas para los objetos de origen de datos para los que se han definido apodos a fin de optimizar las consultas que implican estos apodos. Estas estadísticas se recuperan del origen de datos cuando crea un apodo para un objeto de origen de datos utilizando la sentencia CREATE NICKNAME. La base de datos federada verifica la presencia del objeto en el origen de datos y, a continuación, intenta recopilar los datos estadísticos existentes del origen de datos. La información útil para el optimizador se lee de los catálogos de origen de datos y se pone en el catálogo global en el servidor federado. Debido a que parte o toda la información de catálogo de origen de datos puede ser utilizada por el optimizador, es aconsejable actualizar las estadísticas (utilizando el mandato del origen de datos equivalente a RUNSTATS) en el origen de datos antes de crear un apodo.

Las estadísticas de catálogo describen el tamaño global de las tablas y las vistas, así como el rango de valores en las columnas asociadas. La información recuperada, incluye, pero no se limita a, la siguiente:

- El número de filas en un objeto de apodo
- El número de páginas que ocupa un apodo
- El número de valores distintos en cada columna de una tabla
- El número de valores distintos en las columnas de un índice
- Los valores más altos/más bajos de una columna

Mientras que la base de datos federada puede recuperar los datos estadísticos contenidos en un origen de datos, no puede detectar automáticamente actualizaciones a datos estadísticos existentes en los orígenes de datos. Además, la

base de datos federada no tiene ningún mecanismo para manejar cambios en la definición de objeto o cambios estructurales realizados a objetos en los orígenes de datos (tal como cuando se añade una columna a una tabla).

Si cambian los datos estadísticos o las características estructurales para un objeto remoto en el cual se define un apodo, tiene las opciones siguientes para actualizar las estadísticas:

- Recogida manual de estadísticas
 - Ejecute el equivalente de RUNSTATS en el origen de datos. A continuación, descarte el apodo actual y cree de nuevo el apodo. Éste es el método recomendado para actualizar las estadísticas.
Una ventaja de este método es que además de estadísticas actualizadas, cualquier información sobre nuevos índices o cambios estructurales realizados al objeto remoto se reflejará en el nuevo apodo. Una desventaja de este método es que las vistas o los paquetes basados en el apodo antiguo se invalidarán.
 - Utilice el recurso de actualización de las estadísticas de apodos en el Centro de control de DB2. De forma alternativa, utilice el procedimiento almacenado SYSPROC.NNSTAT() subyacente, disponible desde el procesador de la línea de mandatos.
El recurso de actualización de las estadísticas de apodos (o SYSPROC.NNSTAT()) sólo actualiza las estadísticas de apodos; no altera el apodo para que coincida con ningún cambio estructural en el objeto remoto. Por ejemplo, si el objeto remoto tiene una nueva columna, la actualización de las estadísticas de apodos no añade una columna al apodo.
 - Actualice manualmente las estadísticas en la vista de catálogo SYSSTAT.TABLES. Utilice este método sólo cuando sepa que la información de estadística en el origen de datos remoto es incorrecta o está incompleta.
- Recogida automática de estadísticas
Esta característica se ejecuta por omisión para mejorar el rendimiento mediante la recogida automática de estadísticas actualizadas de tablas y apodos.

Actualización de los cambios de fila

Si se añade una gran cantidad de filas a un objeto en el origen de datos, o si se suprimen una gran cantidad de filas de un objeto en el origen de datos, la base de datos federada no es consciente de estos cambios porque las estadísticas de catálogo para el apodo continúan indicando el número antiguo de filas.

Sin embargo, es posible que note un detrimento en el rendimiento porque el optimizador continúa tomando decisiones en base a la información de estadísticas de apodo que ya no son precisas. Después de actualizar las estadísticas para el objeto remoto en el origen de datos, puede actualizar las estadísticas para el apodo para garantizar que el optimizador pueda utilizar las estadísticas precisas cuando genere y elija planes de acceso para procesar consultas en el origen de datos.

Actualización de las estadísticas cuando cambian las columnas

Cuando hay cambios estructurales a un objeto de origen de datos, por ejemplo, cuando se añade una columna a una tabla, debe completar varios pasos para actualizar las estadísticas para ese objeto en el catálogo de bases de datos federadas.

Acerca de esta tarea

Si se añaden, suprimen o alteran columnas en el origen de datos, es posible que perciba resultados incorrectos o reciba un mensaje de error. Por ejemplo, suponga que el apodo *EUROSALES* hace referencia a la tabla *europa* en una base de datos Sybase. Si se añade una nueva columna denominada *CZECH* a la tabla, la base de datos federada no reconocerá la columna *CZECH*. Las consultas que hacen referencia a esa columna darán como resultado un mensaje de error.

Procedimiento

Para actualizar las estadísticas para ese objeto cuando se produzcan cambios de columnas:

1. Ejecute el programa de utilidad sobre el origen de datos que es equivalente a DB2 RUNSTATS. Esto actualizará las estadísticas almacenadas en el catálogo de origen de datos.
2. Descarte el apodo actual para el objeto de origen de datos utilizando la sentencia DROP NICKNAME.
3. Vuelva a crear el apodo utilizando la sentencia CREATE NICKNAME.

Análisis de la optimización global

La información detallada sobre los planes de acceso, incluyendo parte de la información que utiliza el optimizador global para elegir el plan óptimo, se mantiene en las tablas de Explain separada del propio plan de acceso real.

Esta información permite un análisis en profundidad de un plan de acceso. Las tablas de Explain son accesibles en todos los sistemas operativos soportados y contienen información tanto sobre sentencias de SQL estáticas como sobre sentencias de SQL dinámicas. Puede acceder a las tablas de Explain utilizando sentencias de SQL. Esto permite una fácil manipulación de la salida, para realizar comparaciones entre distintas consultas o para realizar comparaciones de la misma consulta a lo largo del tiempo.

Existen varias maneras de obtener información de plan de acceso global de las tablas de Explain:

- Puede utilizar la herramienta de formato de tablas de Explain, db2exfmt, para presentar la información de las tablas de Explain en un formato predefinido.
- También puede utilizar el recurso de Explain integrado del Centro de control de DB2 conjuntamente con Visual Explain para comprender el plan de acceso elegido para una sentencia de SQL determinada.

Tanto las sentencias de SQL estáticas como las sentencias de SQL dinámicas se explican utilizando el recurso de Explain. Una diferencia entre Visual Explain y db2exfmt es que Visual Explain presenta información en un formato gráfico mientras que db2exfmt presenta la información en formato de texto. Los dos métodos proporcionan el mismo nivel de detalle.

- Puede utilizar las herramientas de db2expln y dynexpln para comprender el plan de acceso elegido para una sentencia de SQL determinada.

Para entender completamente la salida de db2exfmt, Visual Explain, db2expln o dynexpln, debe comprender lo siguiente:

- Las distintas sentencias de SQL soportadas y la terminología relativa a estas sentencias (tal como predicados en una sentencia SELECT)
- El propósito de un paquete (plan de acceso)
- El propósito y el contenido de las tablas de catálogo del sistema

- Los operadores de proceso de consultas básicos, tal como uniones, agrupar por, agregación y ordenaciones

Interpretación de decisiones de optimización de planes de acceso

Esta sección lista cuestiones de optimización típicas y áreas que se pueden investigar para mejorar el rendimiento.

Por qué no se evalúa remotamente una unión entre dos apodos del mismo origen de datos

Puede comprobar elementos de la operación de unión, predicados de unión y el número de filas en el resultado para evaluar por qué una unión entre dos apodos del mismo origen de datos no se evalúa de forma remota.

Las áreas a examinar incluyen:

- Operaciones de unión. ¿Puede el origen de datos dar soporte a una unión?
- Predicados de unión. ¿Se puede evaluar el predicado de unión en el origen de datos remoto?
- Número de filas en el resultado de unión. Puede determinar el número de filas con Visual Explain. ¿Produce la unión un conjunto de filas mucho mayor que los dos apodos combinados? ¿Los números coinciden con la realidad? Si la respuesta es no, considere actualizar las estadísticas de apodo con el procedimiento almacenado SYSPROC.NNSTAT().

Por qué no se evalúa el operador GROUP BY de forma remota

Las áreas a examinar incluyen:

- Sintaxis del operador. Verifique que el operador se puede evaluar en el origen de datos remoto.
- Número de filas. Compruebe el número estimado de filas en la entrada y salida del operador GROUP BY utilizando Visual Explain. ¿Son estos dos números muy próximos? Si la respuesta es sí, el optimizador considera más eficaz evaluar este GROUP BY localmente. Además, ¿reflejan estos dos números la realidad? Si la respuesta es no, considere actualizar las estadísticas de apodo utilizando el procedimiento almacenado SYSPROC.NNSTAT().

Por qué no se evalúa completamente la sentencia de forma remota

El servidor federado busca asegurarse que la semántica de la consulta y los resultados obtenidos para las consultas federadas sean exactamente los mismos que si hubieran sido evaluados completamente por DB2 Database para Linux, UNIX, y Windows. La fase de análisis de envío del compilador de consultas decide si el proceso de envío a orígenes remotos mantendrá la semántica de DB2. Por lo tanto, las operaciones de consultas federadas se pueden enviar de forma segura sólo si las operaciones correspondientes en el origen remoto tienen el mismo significado y resultado. El motivo más habitual de anomalía en la finalización del proceso de envío de una consulta a un único origen remoto es la existencia de leves diferencias en funcionalidad entre el servidor federado y el origen remoto para una o más operaciones en la consulta.

El optimizador realiza optimización basada en costes. Incluso si el análisis de envío indica que cada operador se puede evaluar en el origen de datos remoto, el optimizador todavía confía en su cálculo de costes para generar un plan óptimo

globalmente. Existen muchos factores que contribuyen a la decisión de seleccionar ese plan. Suponga que el origen de datos remoto puede procesar cada operación en la consulta original. Sin embargo, su velocidad de CPU es mucho más lenta que la velocidad de CPU del servidor federado. Puede que resulte más beneficioso realizar en su lugar las operaciones en el servidor federado. Si no se consigue el rendimiento que desea, verifique las estadísticas del servidor en la tabla de catálogo SYSSTAT.SERVEROPTIONS.

Por qué un plan generado por el optimizador y completamente evaluado de forma remota tiene un rendimiento mucho peor que la consulta original ejecutada directamente en el origen de datos remoto

Las áreas a examinar incluyen:

- La sentencia de SQL remota generada por el optimizador de consultas. Además de la sustitución de los apodos por los nombres de tabla remota correspondientes, la sentencia de SQL remota normalmente difiere de la sentencia federada original de las formas siguientes:
 - Es posible que la ordenación de predicados en la consulta haya cambiado.
 - Se han eliminado los predicados encontrados en la consulta original, sustituidos por predicados equivalentes o aumentados por predicados adicionales.
 - Es posible que las subconsultas se hayan sobrescrito como uniones.
 - Se posible que se hayan añadido funciones adicionales que realicen conversión o truncamiento de series para mantener la semántica de DB2

Con la excepción del último elemento listado anteriormente, estos cambios normalmente tienen un impacto favorable sobre el rendimiento. Sin embargo, en unos cuantos casos, es posible que los cambios hagan que el optimizador de consultas remotas genere un plan distinto (y más lento) que el que tendría para la consulta original

Un buen optimizador de consultas no debe ser sensible a la ordenación de predicados de una consulta. Desafortunadamente, no todos los optimizadores de DBMS son idénticos. Es probable que el optimizador en el origen de datos remoto genere un plan distinto en base a la ordenación de predicados de entrada. Si esto es cierto, se trata de un problema inherente en el optimizador remoto. Considere modificar la ordenación de predicados o ponerse en contacto con la organización del servicio del origen de datos remoto para obtener asistencia.

Además, compruebe si hay sustituciones de predicados. Un buen optimizador de consultas no debe ser sensible a las sustituciones de predicados equivalentes. Es posible que el optimizador en el origen de datos remoto genere un plan distinto en base al predicado de entrada. Por ejemplo, algunos optimizadores no pueden generar sentencias de cierre transitivas para los predicados.

- El número de filas devueltas. Puede obtener este número de Visual Explain. Si la consulta devuelve una gran cantidad de filas, el tráfico de red es un cuello de botella potencial.
- Funciones adicionales. ¿Contiene la sentencia de SQL remota más funciones que la consulta original? Es posible que algunas de las funciones adicionales se hayan generado para convertir tipos de datos. Asegúrese de que sean necesarias.

Capítulo 27. Elementos del supervisor del sistema que afectan al rendimiento

El supervisor del sistema de bases de datos federadas recopila información estadística relativa al estado actual del gestor de bases de datos e información de actividad tal como contadores y otras mediciones del proceso de base de datos.

En un sistema federado, puede utilizar el supervisor del sistema de bases de datos para recopilar información sobre la actividad de base de datos, rendimiento del sistema y rendimiento de la aplicación.

El conmutador de supervisor de indicación de fecha y hora se utiliza para realizar un seguimiento de los tiempos de respuesta de las interacciones que la base de datos federada tiene con un origen de datos. Los elementos de datos federados de los que realiza un seguimiento el conmutador de indicación de fecha y hora son los siguientes:

- Crear tiempo de respuesta de apodo
- Suprimir tiempo de respuesta
- Insertar tiempo de respuesta
- Tiempo de paso a través
- Tiempo de respuesta de consulta
- Tiempo de bloqueo remoto
- Tiempo de procedimiento almacenado
- Actualizar tiempo de respuesta

El valor por omisión para el conmutador de supervisor de indicación de fecha y hora es ON (activado).

Recomendación: puede aumentar el rendimiento cambiando el valor del conmutador del supervisor de indicación de fecha y hora a OFF (desactivado) para todas las aplicaciones. Si una aplicación tiene un conmutador de indicación de fecha y hora establecido en ON, el sistema continuará recopilando los tiempos de respuesta. Por lo tanto, no aumentará el rendimiento desactivando el conmutador de indicación de fecha y hora para sólo algunas de las aplicaciones.

La desactivación del conmutador tiene otras implicaciones.

- La desactivación del supervisor de indicación de fecha y hora para todas las aplicaciones requiere que detenga y reinicie la instancia de DB2 para implementar el cambio.
- La desactivación del supervisor de indicación de fecha y hora inhabilita la recopilación de información de indicación de fecha y hora tanto en las aplicaciones federadas como en las no federadas. La base de datos local no recibirá tampoco información de indicación de fecha y hora.

Si necesita información de indicación de fecha y hora para aplicaciones no federadas locales, no debe desactivar el conmutador de supervisor de indicación de fecha y hora.

Puede establecer el conmutador de indicación de fecha y hora en OFF para todas las aplicaciones utilizando este mandato:

```
update dbm cfg using dft_mon_timestamp off
```

A continuación, emita:

```
db2stop  
db2start
```

La detención y el inicio del servidor federado garantizarán que el conmutador esté desactivado para todas las aplicaciones.

La información específica para cada una de los elementos de los que el conmutador de indicación de fecha y hora realiza un seguimiento se trata en un tema distinto.

Capítulo 28. Tablas de consultas materializadas

Una tabla de consultas materializadas es una tabla que coloca en memoria caché los resultados de una consulta. Cuando se envía de nuevo la consulta, el motor de la base de datos puede devolver los datos desde la tabla de consultas materializadas. Puede utilizar tablas de consultas materializadas con apodos para mejorar el rendimiento de una consulta.

Las tablas de consultas materializadas se utilizan al crear una tabla de memoria caché. La tabla de memoria caché almacena datos locales definidos por las tablas de consultas materializadas que están asociadas a la tabla.

Tablas de consultas materializadas y sistemas federados - Visión general

Una tabla de consultas materializadas es una tabla que coloca en memoria caché los resultados de una consulta. Cuando se envía la consulta de nuevo, el motor de la base de datos puede devolver los datos desde la tabla de consultas materializadas en lugar de repetir el cálculo de la consulta.

Se pueden utilizar tablas de consultas materializadas con apodos para mejorar el rendimiento de una consulta y para encapsular una parte de lógica. Las tablas de consultas materializadas se utilizan al crear tablas de memoria caché.

El optimizador de SQL determina si una consulta se ejecutará más eficazmente con una tabla de consultas materializadas que las tablas base o los apodos. El optimizador utiliza los siguientes factores para seleccionar una tabla de consultas materializadas:

- La tabla de consultas materializadas debe coincidir en parte o totalmente con la consulta.
- Se debe cumplir el criterio de antigüedad de renovación.
- El plan de acceso que utilice una tabla de consultas materializadas debe ser más barato que el plan de acceso que utilice las tablas base o los apodos.

Se da soporte a las tablas de consultas materializadas que implican apodos para objetos de los siguientes orígenes de datos:

- Orígenes de datos relacionales
 - DRDA
 - Informix
 - JDBC
 - ODBC
 - Oracle
 - Sybase
 - Microsoft SQL Server
 - Teradata
- Orígenes de datos no relacionales
 - BioRS
 - Excel
 - Archivos con estructura de tabla
 - Servicios Web
 - XML

Creación de una tabla de consultas materializadas federada

Puede utilizar tablas de consultas materializadas para colocar en memoria caché los datos de forma local y para mejorar el rendimiento de las consultas. Puede utilizar apodos de orígenes de datos relacionales y no relacionales para crear tablas de consultas materializadas.

Restricciones

- “Restricciones específicas de orígenes de datos para tablas de consultas materializadas”
- Si la consulta tiene una plantilla de función en un predicado o una lista de selección, la plantilla de función debe formar parte de la tabla de consultas materializadas.
- “Restricciones sobre la utilización de tablas de consultas materializadas con apodos” en la página 285

Procedimiento

Para crear una tabla de consultas materializadas, emita una sentencia CREATE TABLE que haga referencia a los apodos que representan los objetos de origen de datos remota que desea incluir.

Puede llenar una tabla de consultas materializadas mantenida por el usuario utilizando una sentencia INSERT en una sentencia de subselección. Por ejemplo:

```
insert into my_mqt (select ..from n1, n2 ..)
```

donde la parte select de la consulta coincide con la definición de la tabla de consultas materializadas. Es posible que el optimizador utilice my_mqt para sustituir la parte select de la consulta. En este caso, la sentencia se convierte en:

```
insert into my_mqt (select .. from my_mqt);
```

En este caso, la tabla de consultas materializadas se convierte en el fuente de la operación insert. Para evitar que esto suceda, puede emitir uno de los mandatos siguientes para inhabilitar temporalmente la tabla de consultas materializadas:

```
SET CURRENT REFRESH AGE 0  
SET CURRENT MAINTAINED TABLE TYPE FOR OPTIMIZATION SYSTEM
```

Restricciones específicas de orígenes de datos para tablas de consultas materializadas

Cuando crea tablas de consultas materializadas, necesita conocer las restricciones para orígenes de datos específicos.

En este tema se describen las restricciones de la creación de tablas de consultas materializadas para los siguientes orígenes de datos:

- BioRS
- Archivos con estructura de tabla
- Servicios Web
- XML

Restricciones de búsqueda de BioRS

El derivador de BioRS requiere como mínimo un predicado en la cláusula WHERE. Debe crear una tabla de consultas materializadas que satisfaga los requisitos de predicado del derivador. Si no especifica un predicado, una renovación de una tabla de consultas materializadas falla.

Restricciones de archivos estructurados por tabla

Si define un apodo para un archivo estructurado por tablas con la opción DOCUMENT, la tabla de consultas materializadas debe tener un predicado que especifique la vía de acceso del archivo. Si no especifica un predicado, una renovación de una tabla de consultas materializadas falla.

Restricciones de servicios Web

Sólo puede crear una tabla de consultas materializadas a través de una vista aplanada de una jerarquía de apodos. No puede crear una tabla de consultas materializadas para cada apodo de una jerarquía.

Restricciones de XML

No puede crear una tabla materializada en una tabla hijo.

Si define un apodo para una tabla XML con la opción DOCUMENT, la tabla de consultas materializadas requiere un predicado que especifique la vía de acceso del archivo. Si no especifica un predicado, una renovación de una tabla de consultas materializadas falla.

Restricciones sobre la utilización de tablas de consultas materializadas con apodos

Al ajustar el sistema federado, tenga en cuenta se deben considerar estas restricciones sobre las tablas de consultas materializadas que hacen referencia a apodos.

Control de acceso basado en etiquetas (LBAC) para objetos de origen de datos

Los apodos para objetos de origen de datos que utilizan control de acceso basado en etiquetas o seguridad de etiquetas de Oracle no se pueden colocar en memoria caché, y no se pueden crear tablas de consultas materializadas sobre los mismos.

Tablas de consultas materializadas mantenidas por el sistema

El sistema federado no da soporte a las tablas de consultas materializadas mantenidas por el sistema que hacen referencia a apodos en un entorno de base de datos particionado.

El derivador federado debe estar limitado a fin de que el recurso de reescritura federada direcciona la consulta a la MQT. Puede crear el derivador como FENCED o bien cambiar el derivador utilizando la sentencia ALTER WRAPPER. Por ejemplo:

```
CREATE WRAPPER <nombre de derivador>
  LIBRARY <nombre_bib1>
  OPTIONS (DB2_FENCED 'N');
ALTER WRAPPER <nombre de derivador> OPTIONS (SET DB2_FENCED 'Y');
```

Necesita establecer los tipos de tablas mantenidas para la optimización en ALL o USER utilizando el parámetro de configuración de base de datos DFT_MTTB_TYPES o bien el registro especial actual set emitido antes del SQL.

Para cambiar el parámetro de configuración al valor USER, emita lo siguiente:

```
update db cfg for <alias_bd> using DFT_MTTB_TYPES USER
```

Para utilizar el registro especial set, emita lo siguiente:

```
set current maintained table types for optimization ALL
```

Para solucionar temporalmente esta restricción, puede utilizar tablas de consultas materializadas definidas por el usuario.

Por ejemplo, para un apodo no relacional denominado DEPART, puede emitir los mandatos siguientes para simular una tabla de consultas materializadas mantenida por el sistema.

```
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;
```

La siguiente sentencia SELECT puede ser contestada por la tabla de consultas materializadas definida anteriormente:

```
SELECT EMPNO, FIRSTNME FROM DEPART  
WHERE EMPNO > '000000' AND FIRSTNME LIKE 'AN%';
```

Capítulo 29. Tablas de memoria caché

Puede utilizar las tablas de memoria caché para almacenar los datos a los que accede con frecuencia pero que no cambian a menudo.

Una tabla de memoria caché puede mejorar el rendimiento de las consultas almacenando los datos localmente en lugar de acceder a los datos directamente desde el origen de datos.

Puede almacenar datos en la memoria caché de estos orígenes de datos:

- Familia de productos DB2
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

Una tabla de memoria caché está formada por estos componentes:

- Un apodo del sistema de bases de datos federado. El apodo tiene las mismas definiciones de columna y los mismos accesos a datos que la tabla de origen de datos.
- Una o más tablas de consulta materializadas que se definen en el apodo. El tipo de tablas de consulta materializadas es una tabla de consulta materializada y mantenidas FEDERATED_TOOL. La tabla de consulta materializada generalmente contiene un subconjunto de datos que se utilizan muy a menudo de la tabla de origen de datos.
- Una planificación de réplicas para la tabla de consultas materializada. La planificación de réplicas mantiene las tablas de consulta materializadas actuales con las tablas del origen de datos. Debe definir la planificación de réplicas.

En la figura siguiente se ilustra una tabla de memoria caché.

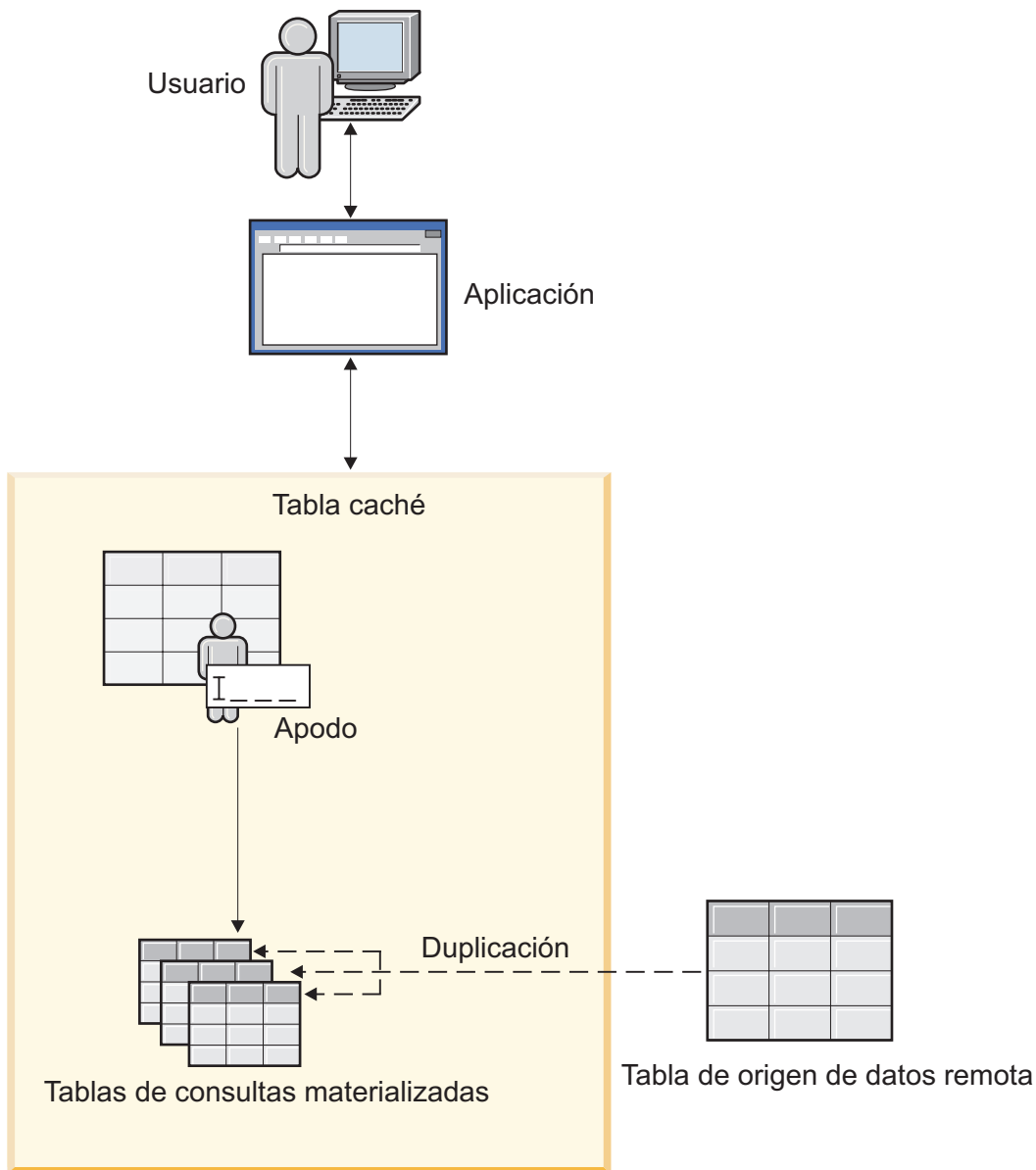


Figura 14. Tabla de memoria caché

La tabla de memoria caché tiene el mismo nombre que el apodo. Puede asociar una tabla de memoria caché sólo con una tabla de origen de datos.

Cuando una tabla de memoria caché está habilitada, el optimizador de consultas direcciona las consultas a la tabla de memoria caché si los datos solicitados por la consulta se encuentran en la tabla de consulta materializada.

Creación de tablas de memoria caché

Para crear una tabla de memoria caché, se utiliza un asistente del Centro de control. El asistente crea el apodo, la tabla de consulta materializada y la planificación de réplicas necesarias para la tabla de memoria caché.

Antes de empezar

- Establezca el parámetro FEDERATED en YES en el servidor federado. El parámetro FEDERATED es un parámetro de configuración del gestor de bases de datos.
- Para acceder a orígenes de datos Informix, instale y configure el kit de desarrollo de software (SDK) del cliente Informix en el servidor federado.
- Para almacenar datos en memoria caché desde tablas de DB2 Database for Linux, UNIX, and Windows, configure la base de datos DB2 para registro de archivado.
- La base de datos federada o la base de datos origen deben encontrarse en el sistema desde el que se crean las tablas de memoria caché. Si la base de datos federada o la base de datos origen no son locales, debe catalogar las bases de datos en el sistema local. El nombre de alias utilizado al catalogar la base de datos debe ser el mismo que el de la base de datos.
- El ID de usuario de la correlación de usuarios entre las bases de datos debe tener autoridad para crear tablas en la base de datos origen.

Procedimiento

Para crear una tabla de memoria caché:

1. En el Centro de control, expanda la carpeta **Objetos de memoria caché**.
2. Pulse la carpeta **Tablas de memoria caché** con el botón derecho del ratón y, a continuación, pulse **Crear**.
3. Siga los pasos del asistente Tabla de memoria caché para crear la tabla de memoria caché. Puede crear la tabla de memoria caché rápidamente especificando sólo valores para los campos obligatorios y utilizando valores predeterminados para el resto de campos. Para cambiar los valores predeterminados:
 - a. En la página Tabla de consulta materializada, pulse **Valores avanzados de MQT** para cambiar los valores predeterminados o para seleccionar un subconjunto de columnas para la tabla de consulta materializada.
 - b. En la página Réplica, pulse **Valores avanzados** para cambiar los valores predeterminados para replicar datos de la tabla del origen de datos en la tabla de consulta materializada.

En algunos casos, al almacenamiento en memoria caché no se habilitará cuando finalice el asistente. Deberá habilitar la memoria caché para iniciar los programas de captura y aplicación de réplica.

El asistente Tabla de memoria caché crea una tabla de consulta materializada cuando el usuario crea la tabla de memoria caché. Puede crear tablas de consulta materializadas adicionales para almacenar otros datos del mismo origen de datos.

Modificación de valores para tablas de consulta materializadas

No es posible modificar directamente los valores para las tablas de consulta materializadas. Debe utilizar métodos alternativos para cambiar los valores de réplica y de las tablas de consulta materializadas.

Procedimiento

Para modificar los valores para una tabla de consulta materializada:

1. Utilice la ventana Detalles de tabla de consulta materializada para ver los valores para la tabla de consulta materializada y la planificación de réplicas.
 - a. En el Centro de control, expanda la carpeta **Objetos de memoria caché**.

- b. Pulse con el botón derecho en la tabla de memoria caché y seleccione **Propiedades**.
 - c. Seleccione la tabla de consulta materializada y pulse **Detalles** para ver los valores actuales.
2. Si debe realizar cambios en los valores de réplica, utilice el centro de réplica. No puede cambiar los valores de réplica para la tabla de consulta materializada desde la ventana Detalles de tabla de consulta materializada.
3. Si debe cambiar los valores para una tabla de consulta materializada, elimine la tabla de consulta materializada y cree otra tabla de consulta materializada. Por ejemplo, si debe añadir otra columna a la tabla de consulta materializada, elimine la tabla de consulta materializada y cree una tabla de consulta materializada con los nuevos valores.

Adición de tablas de consulta materializadas a una tabla de memoria caché

Puede crear tablas de consulta materializadas adicionales para la misma tabla de memoria caché. Utilice las tablas de consulta materializadas adicionales para almacenar otros datos de la misma tabla de origen de datos.

Acerca de esta tarea

Cuando se crea una tabla de memoria caché, el servidor federado almacena datos del origen de datos localmente en una tabla de consulta materializada. Los criterios que especifique en el asistente Tabla de memoria caché determinarán los datos que se almacenan en la tabla de consulta materializada.

Por ejemplo, tiene una tabla de consulta materializada inicial que contiene información acerca de clientes que se encuentran en Asia. Puede crear otra tabla de consulta materializada que contenga información acerca de clientes que se encuentran en Sudamérica.

Procedimiento

Para añadir una tabla de consulta materializada a una tabla de memoria caché existente:

1. En el Centro de control, expanda la carpeta **Objetos de memoria caché** y la carpeta **Tablas de memoria caché**.
2. Pulse la carpeta adecuada con el botón derecho del ratón y pulse **Propiedades**.
3. Pulse **Añadir**.
4. Siga los pasos del asistente Tabla de memoria caché para crear la tabla de consulta materializada adicional.

También puede acceder a la ventana Propiedades de tabla de memoria caché desde un objeto de apodo. Pulse el apodo con el botón derecho del ratón y pulse **Memoria caché**.

Direccionamiento de consultas a tablas de memoria caché

Puede direccionar consultas a la tabla de consulta materializada para la tabla de memoria caché o el origen de datos direccionando las consultas al apodo.

Antes de empezar

- Debe existir una tabla de memoria caché para el apodo que se consulta.
- Habilite el almacenamiento en memoria caché para la tabla de consulta materializada.
- Establezca los siguientes parámetros de configuración de la base de datos:
 - Tipo de tabla materializada para optimización (DFT_MTTB_TYPES)
 - Clase de optimización de consultas (DFT_QUERYOPT)

Utilice el cuaderno **Configurar base de datos** del Centro de control o la línea de mandatos para establecer estos parámetros.

Procedimiento

Para cambiar el direccionamiento para las consultas:

1. En el Centro de control, expanda las carpetas **Objetos de memoria caché** y **Tablas de memoria caché**.
2. Pulse con el botón derecho en la tabla de memoria caché adecuada y seleccione **Propiedades**.
3. Seleccione la tabla de consulta materializada y pulse **Comprobar estado**.
4. Seleccione si desea que las consultas direccionen las consultas:
 - Seleccione **Direccionar a tabla de consulta materializada**. Todas las consultas para el origen de datos se envían a la tabla de consulta materializada. Si los datos de la tabla de consulta materializada no satisfacen la consulta, la consulta se direcciona al origen de datos utilizando el apodo.
 - Seleccione **Direccionar a apodo**. Todas las consultas se envían al origen de datos utilizando el apodo. La tabla de consulta materializada cambia por una tabla de base de datos normal. La réplica a la tabla de consulta materializada continúa a menos que se inhabilite el almacenamiento en la memoria caché.
5. Para cambiar de forma inmediata el direccionamiento, active el recuadro de selección **Eliminar todas las sentencias SQL dinámicas en memoria caché**. Las sentencias SQL dinámicas que se encuentran en la memoria caché del paquete se eliminan.
6. Pulse **Aceptar**.

También puede acceder a la ventana Propiedades de la tabla de memoria caché desde un objeto de apodo. Pulse con el botón derecho en el apodo y seleccione **Almacenamiento en memoria caché**.

Habilitación e inhabilitación de los valores de memoria caché de réplica

La réplica de datos para la tabla de consulta materializada se inicia y detiene cambiando los valores de memoria caché.

Antes de empezar

Para orígenes de datos DB2 Database para Linux, UNIX y Windows, establezca el tipo de registro de bases de datos en el registro de archivado.

Acerca de esta tarea

Al habilitar el almacenamiento en la memoria caché, los programas Capturar y Aplicar se inician si todavía no están ejecutándose. Al mismo tiempo, se habilita el miembro del conjunto de suscripción. La habilitación del miembro del conjunto de suscripción indica al programa Aplicar que mantenga los datos de la tabla de consulta materializada sincronizados con los datos de la tabla de origen de datos.

Cuando se inhabilita el almacenamiento en memoria caché, el miembro del conjunto de suscripción se inhabilita. Los datos del origen de datos no se replican en la tabla de consulta materializada.

Importante: Si no cambia el valor de direccionamiento, las consultas se dirigen a la tabla de consultas materializadas aun cuando los datos no se repliquen en la tabla de consulta materializada.

Procedimiento

Para habilitar o inhabilitar los valores de memoria caché de réplica para una tabla de consulta materializada:

1. En el Centro de control, expanda las carpetas **Objetos de memoria caché** y **Tablas de memoria caché**.
2. Pulse con el botón derecho en la tabla de memoria caché adecuada y seleccione **Propiedades**.
3. Seleccione la tabla de consulta materializada y pulse **Comprobar estado**.
4. Seleccione **Habilitar almacenamiento en memoria caché** o **Inhabilitar almacenamiento en memoria caché**. Para ver los mandatos de habilitación o inhabilitación, pulse **Mostrar sentencias**.
5. Pulse **Aceptar**.

También puede acceder a la ventana **Propiedades de la tabla de memoria caché** desde un objeto de apodo. Pulse con el botón derecho en el apodo y seleccione **Almacenamiento en memoria caché**.

Eliminación de tablas de consulta materializadas de una tabla de memoria caché

Cuando ya no desee almacenar datos localmente en una tabla de consulta materializada, puede eliminarla de la tabla de memoria caché.

Acerca de esta tarea

Si una tabla de memoria caché sólo tiene una tabla de consulta materializada, al eliminar ésta también se eliminará la tabla de memoria caché.

Para asegurarse de que la tabla de consulta materializada se ha eliminado completamente del sistema, utilice el Centro de control para eliminar la tabla de consulta materializada de la tabla de memoria caché.

Procedimiento

Para eliminar una tabla de consulta materializada de una tabla de memoria caché:

1. En el Centro de control, expanda la carpeta **Objetos de memoria caché** y la carpeta **Tablas de memoria caché** en el árbol de objetos.

2. Pulse la carpeta adecuada con el botón derecho del ratón y pulse **Propiedades**.
3. Seleccione la tabla de consulta materializada adecuada y pulse **Eliminar**.

Eliminación de tablas de memoria caché

Cuando ya no desee almacenar datos localmente en una tabla de memoria caché, puede eliminarla.

Acerca de esta tarea

Cuando se elimina una tabla de memoria caché, el servidor federado realiza las acciones siguientes:

- Elimina las tablas de consulta materializadas correspondientes a la tabla de memoria caché.
- Elimina la planificación de réplicas para los orígenes de datos y las tablas de consulta materializadas.
- Elimina el apodo del origen de datos, si si apodo se ha creado al crear la tabla de memoria caché. Si ha utilizado un apodo existente al crear la tabla de memoria caché, el servidor federado no lo elimina.

Procedimiento

Para eliminar una tabla de memoria caché:

1. En el Centro de control, expanda la carpeta **Objetos de memoria caché** y la carpeta **Tablas de memoria caché** en el árbol de objetos.
2. Pulse la tabla de memoria caché en cuestión con el botón derecho del ratón y pulse **Eliminar**.

Capítulo 30. Seguridad para servidores federados

El servidor federado soporta la capa de sockets seguros (SSL) para el cifrado de datos, así como los proxy HTTP y SOCKS para orígenes de datos específicos.

Cifrado

El cifrado proporciona un nivel de seguridad superior al proporcionado por un nombre y una contraseña. El estándar de Internet para el cifrado entre puntos finales es SSL (capa de sockets seguros) y TLS (seguridad de la capa de transporte). SSL utiliza certificados firmados para asegurar la seguridad de las comunicaciones. Un *certificado* es un documento digital que proporciona seguridad a la identidad del usuario o del servidor. Una entidad emisora de certificados, como VeriSign, firma el certificado, aunque también puede autofirmarse. Los socios de comunicaciones determinan si se debe o no aceptar un certificado concreto como auténtico.

Los socios tiene un almacén de certificados o almacén de claves. El almacén de claves conserva los certificados que el socio acepta de terceros y certificados que le representan. Cada punto final determina si se debe o no enviar un certificado al abrir las comunicaciones y si se deben o no aceptar las comunicaciones de un socio que no proporcione un certificado.

Para los derivadores y las funciones, las funciones SSL siguientes son importantes:

- La identificación de lado del servidor de un certificado que se debe enviar
- La verificación del lado del servidor de un cliente que se debe certificar
- La verificación del lado del cliente de un servidor certificado
- La identificación de lado del cliente de un certificado que se debe enviar
- Tipo de soporte, ubicación y acceso del cliente y del servidor a un almacén de claves local.

La interacción entre SSL y un proxy depende del tipo de proxy. En general, las comunicaciones SSL se transmiten a través de un proxy. La sesión del proxy se establece en texto simple.

IBM Global Security Kit (GSKit) proporciona servicios de cifrado para derivadores y funciones definidas por el usuario.

Proxies

Para impedir ataques desde Internet, muchas empresas aplican cortafuegos. Un *cortafuegos* es una configuración de red que se compone de hardware y software y que suele ubicarse en un límite de comunicación, por ejemplo entre una intranet de la empresa e Internet. El cortafuegos actúa como guardián para regular el tráfico en el límite. En la mayoría de los casos, el cortafuegos impide que las comunicaciones no deseadas crucen el límite; pero, en ocasiones puede bloquear comunicaciones legítimas.

Para garantizar que todas las comunicaciones legítimas pasen por el cortafuegos, se implementa un proxy. Un *proxy* es un programa de servidor autorizado para comunicar a través del cortafuegos. Cuando un programa necesita conectarse a un servidor remoto, el usuario realiza una solicitud al proxy, que se conecta al

servidor remoto. Después de haber establecido la conexión, el proxy controla el tráfico entre el programa del usuario y el servidor remoto. Este proceso asegura que el programa del usuario pueda cruzar el cortafuegos y refuerza la seguridad: el servidor remoto sólo conoce la dirección del proxy, no la del programa del usuario.

Los servidores proxy SOCKS y HTTP controlan el tráfico entre programas del usuario y servidores remotos. Un proxy SOCKS opera en la capa de transporte y transmite mensajes de transporte arbitrarios (TCP o UDP) entre dos direcciones. El servidor federado soporta SOCKS4 y SOCKS5. SOCKS4, que soporta IPv4, no soporta la autenticación del usuario. Por tanto, cualquiera puede comunicarse a través de un proxy SOCKS4 sin la necesidad de proporcionar las credenciales de usuario. SOCKS5, que soporta IPv6, soporta varias modalidades de autenticación de usuarios. Internet Engineering Task Force (IETF) ha aprobado SOCKS5 como estándar. Para obtener más información, visite www.ietf.org y consulte RFC1928, RFC1929 y RFC1961.

Para utilizar un proxy SOCKS, la capa de transporte se debe configurar, de antemano, para utilizar un proxy. Después de haber abierto una conexión con el proxy, la capa de transporte solicita que el proxy abra una conexión con el servidor remoto. Si se ha configurado para solicitar autenticación, el proxy SOCKS debe solicitar al programa un ID y una contraseña antes de abrir la conexión con el servidor remoto.

Un proxy HTTP trabaja con el protocolo HTTP, que es un protocolo de capa de aplicación. Después de haber abierto una conexión TCP/IP con el proxy, el programa del usuario envía una solicitud que incluye el nombre del servidor remoto. A continuación, el programa del usuario sigue enviando solicitudes a través del servidor proxy. Este proceso cambia ligeramente si el servidor remoto necesita autenticación. Si el servidor remoto necesita autenticación, el servidor proxy envía un mensaje de respuesta que incluye la cabecera de autenticación del proxy. Esta cabecera incluye información sobre el tipo de autenticación que se debe realizar. El programa del usuario reenvía la solicitud e incluye una cabecera de autorización del proxy, que contiene la respuesta a la solicitud de autenticación.

Capítulo 31. Contextos fiables federados y conexiones fiables

Mejore el rendimiento del sistema y minimice o reduzca completamente el uso y el mantenimiento de las correlaciones de usuario.

Un *contexto fiable* es un objeto de base de datos DB2 que define una relación fiable entre un cliente y un origen de datos, por ejemplo, entre un servidor de aplicaciones y un servidor federado o entre un servidor federado y un servidor de base de datos remoto. Para definir una relación fiable, el contexto fiable especifica *atributos fiables*. Existen tres tipos de atributos fiables:

- El ID de autorización del sistema que realiza la solicitud inicial de conexión de base de datos
- La dirección IP o el nombre de dominio desde donde se realiza la conexión
- El valor de cifrado para las comunicaciones de datos entre el servidor de base de datos y el cliente de base de datos

Una *conexión fiable* se establece cuando todos los atributos de una solicitud de conexión coinciden con los atributos fiables especificados en un objeto de contexto fiable definido en el servidor. Después de haber establecido una conexión fiable explícita, los usuarios pueden cambiarse en la misma conexión física, con o sin autenticación. Así mismo, se pueden otorgar roles a los usuarios que especifiquen privilegios para utilizar sólo en la conexión fiable.

Este ejemplo crea un objeto de contexto fiable para 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
```

En este ejemplo, sólo BOSS puede iniciar una conexión fiable desde la dirección IP 9.26.111.111. Mary puede reutilizar la conexión, pero primero debe autenticarse. A continuación, obtendrá el rol adicional MANAGER (gestor), que especifica los privilegios que puede utilizar Mary en la conexión fiable. Otros usuarios, especificados como PUBLIC, pueden reutilizar la conexión sin necesidad de autenticarse. Dichos usuarios pueden obtener el rol adicional AUDITOR, que especifica los privilegios que pueden utilizar en la conexión fiable. Los privilegios adicionales están disponibles para otros usuarios sólo mientras sean usuarios activos de la conexión fiable.

Una conexión fiable puede ser explícita o implícita. El tipo de conexión determina si la conexión se puede reutilizar y si los usuarios pueden obtener roles adicionales.

Una *conexión fiable implícita* se establece cuando una conexión fiable no se solicita explícitamente pero los atributos de conexión coinciden con los atributos fiables de un objetos de contexto fiable en el servidor. Después de haber establecido una conexión fiable implícita, sólo el originador de la conexión fiable puede heredar los roles que, de otra manera, no estarían a su disposición. El resto de usuarios no pueden reutilizar una conexión fiable implícita.

Una *conexión fiable explícita* se establece cuando una aplicación utiliza una API para solicitar una conexión fiable. Si los atributos de la conexión coinciden con los atributos fiables de un contexto fiable, se establecerá una conexión fiable. De lo contrario, se establecerá una conexión normal. Después de haber establecido una conexión fiable explícita, el resto de usuarios podrán reutilizar la conexión y, tanto el originador de la conexión como los usuarios de la misma podrán heredar roles adicionales que, de otra manera, no estarían a su disposición.

Ventajas de las conexiones fiables federadas

En un modelo de aplicación multinivel, las conexiones fiables federadas reutilizan una única conexión física para propagar la identidad real de los usuarios a través de los niveles al servidor de la base de datos.

Para entender las ventajas de las conexiones fiables federadas, se deben tener en cuenta los problemas inherentes al modelo de aplicación con múltiples niveles. un modelo de aplicación con múltiples niveles consta de usuarios empresariales (Nivel 1) que interactúan con una aplicación que se ejecuta en el servidor de aplicaciones (Nivel 2), que direcciona todos los accesos de base de datos a través del servidor federado (Nivel 3), que, a su vez, gestiona las comunicaciones con varios servidores de bases de datos (Nivel 4). En este modelo, el servidor de aplicaciones autentica a los usuarios y gestiona las interacciones con el servidor federado. El servidor federado traduce las solicitudes de usuario en formatos específicos de origen de datos, establece conexiones con los orígenes de datos remotos y les envía solicitudes.

Este modelo utilice el ID y la contraseña de servidor de aplicaciones para crear una conexión con el servidor de base de datos. El servidor federado sólo pasa el ID y la contraseña del servidor de aplicaciones al servidor de base de datos. El servidor de base de datos utiliza los privilegios de base de datos asociados con este ID para autorizar y auditar todas las transacciones que realiza el servidor de aplicaciones, incluidas todas las que realiza en nombre de los usuarios empresariales.

Al utilizar el ID del servidor de aplicaciones se presentan los problemas siguientes:

- La identidad real del usuario que realiza la transacción es desconocida porque el servidor de aplicaciones realiza todas las transacciones.
- Los usuarios no pueden responsabilizarse de las transacciones porque no se les puede auditar.
- El principio del privilegio primero se viola porque el ID del servidor de aplicaciones necesita el superconjunto de todos los privilegios que necesitan todos los usuarios.
- Los datos serán vulnerables si se compromete el ID del servidor de aplicaciones.

Las conexiones fiables federadas abordan estos problemas y proporcionan las ventajas que mejorarán la seguridad y el rendimiento del sistema:

La identidad del usuario es conocida

Puesto que los usuarios están conectados, se conoce la identidad real de los usuarios que acceden a la base de datos.

Se responsabiliza a los usuarios

Los registros de auditoría de la base de datos federada y de la base de datos del origen de datos remotos identifican las transacciones que realiza el servidor de aplicaciones y las que realizan los usuarios individuales. Por tanto, se puede responsabilizar a usuarios concretos de transacciones específicas.

Los privilegios son limitados

Cuando se crea un contexto fiable, se puede otorgar un rol de base de datos predeterminado a todos los usuarios y roles específicos a usuarios específicos. Sólo las conexiones de base de datos fiables que coincidan con la definición del contexto fiable pueden beneficiarse de los privilegios asociados a dicho rol.

Los datos son menos vulnerables

En un sistema que utiliza conexiones fiables federadas, el ID de servidor de aplicaciones no necesita el superconjunto de todos los privilegios que necesitan todos los usuarios. Por tanto, si en algún momento se comprometiera el ID de servidor de aplicaciones, los datos serían menos vulnerables que si el ID tuviera el superconjunto de todos los privilegios que necesitan todos los usuarios.

Se minimiza el mantenimiento administrativo

Se reduce la necesidad de crear y mantener las correlaciones de usuario.

Mejora el rendimiento

Después de establecer una conexión fiable explícita, el servidor federado puede cambiar del ID de usuario actual de la conexión a un ID de usuario distinto, con o sin necesidad de autenticación. La reutilización de la misma conexión física para distintos usuarios mejora el rendimiento.

Tipos de conexiones fiables federadas

Las conexiones fiables federadas son conexiones fiables de extremo a extremo, o bien conexiones fiables de salida. El tipo de conexión que se establezca dependerá del modo en que se configure el sistema y de si la solicitud de conexión de entrada es fiable.

Las configuraciones federadas suelen tener múltiples niveles; es decir, incluyen un servidor de aplicaciones, un servidor federado y un servidor de origen de datos remoto. En esta configuración, el servidor federado recibe las solicitudes de conexión de entrada desde el servidor de aplicaciones y envía las solicitudes de conexión al servidor de origen de datos.

Conexiones fiables federadas de extremo a extremo

Una conexión fiable federada de extremo a extremo proporciona reutilización de conexión y aserción de identidad para las conexiones de entrada y de salida. Por ejemplo, cuando se acredita una conexión de entrada al servidor federado, ya sea explícita o implícitamente, el servidor federado solicita automáticamente una conexión fiable de salida. Si los datos proporcionan una funcionalidad de aserción de identidad, se realizará una conexión de salida fiable y se propagará la identidad del usuario al origen de datos remoto. Las conexiones de entrada y de salida se reutilizan cada vez que un usuario solicita la reutilización de la conexión fiable y la nueva identidad del usuario se propaga por el sistema.

Para los orígenes de datos que no proporcionan una funcionalidad de aserción de identidad, el servidor federado proporciona una aserción de identidad de extremo a extremo pero no proporciona la reutilización de conexión. En estos casos, cada vez que se cambia el usuario, el servidor federado cierra la conexión de salida de los usuarios anteriores y crea una conexión nueva para el usuario nuevo. Al hacerlo, la identidad del usuario se propaga por el sistema incluso si no se reutiliza la conexión de salida.

Conexiones fiables federadas de salida

Una conexión fiable federada de salida aprovecha la capacidad de los orígenes de datos de reutilizar las conexiones fiables sin autenticación para eliminar la necesidad de almacenaje de contraseñas de origen de datos en correlaciones de usuario. Si la conexión de entrada es fiable, el servidor federado solicitará automáticamente una conexión de salida fiable reutilizada sin autenticación. Para las conexiones de entrada no fiables, las conexiones fiables federadas de salida proporcionan la capacidad de reutilizar las conexiones sin necesidad de autenticación de los usuarios.

En esta configuración, utilice la opción `FED_PROXY_USER` en la definición del servidor para especificar el ID de autorización que inicialmente establece la conexión de salida. El ID de autorización especificado debe tener una correlación de usuario que incluya las opciones `REMOTE_AUTHID` y `REMOTE_PASSWORD`.

En función de la configuración del contexto fiable en el servidor de origen de datos remoto, se puede reducir el número de correlaciones de usuario en el catálogo de base de datos hasta uno. Por ejemplo, si se permite que `PUBLIC` se conecte sin autenticación, sólo el usuario de proxy federado necesita correlación de usuario. Sin embargo, si el contexto fiable federado del origen de datos remoto especifica que ciertos usuarios se deben autenticar o si los usuarios no utilizan el mismo ID en el servidor federado y en el origen de datos remoto, se debe crear o alterar la correlación de usuario del usuario para que sólo utilice la opción `REMOTE_AUTHID`, la opción `REMOTE_PASSWORD` o las opciones `REMOTE_AUTHID` y `REMOTE_PASSWORD` y se debe establecer la opción `USE_TRUSTED_CONTEXT` en 'Y'.

Puesto que la conexión fiable federada de salida implica numerosas tareas adicionales de configuración y mantenimiento, diseñe el sistema federado para implementar conexiones fiables de extremo a extremo siempre que sea posible. Utilice conexiones fiables federadas de salida sólo cuando sea absolutamente necesario.

API para conexiones fiables federadas

La API utilizada para solicitar y reutilizar una conexión fiable depende del tipo de aplicación.

Para solicitar una conexión fiable federada, la aplicación debe especificar estas API.

Aplicaciones CLI/ODBC

Utilice `SQLSetConnectAttr` con el atributo `SQL_ATTR_USE_TRUSTED_CONTEXT` para indicar si el cliente solicita una conexión fiable. A continuación, emita un `SQLConnect`.

Aplicaciones XA CLI/ODBC

En una solicitud de serie `xa_open`, establezca el atributo `TCTX` en `True` para solicitar una conexión fiable.

Aplicaciones Java

Utilice `getDB2TrustedPooledConnection` o `getDB2TrustedXAConnection` para solicitar una conexión fiable.

Para reutilizar la conexión para otro usuario, tanto si necesita autenticación como si no la necesita, la aplicación debe especificar estas API:

Aplicaciones CLI/ODBC

Utilice `SQLSetConnectAttr` para establecer `SQL_ATTR_TRUSTED_CONTEXT_USERID` y `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` y especificar el ID de usuario y la contraseña del usuario con el que se está estableciendo la conexión.

Aplicaciones XA CLI/ODBC

Utilice `SQLSetConnectAttr` para establecer `SQL_ATTR_TRUSTED_CONTEXT_USERID` y `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` y especificar el ID de usuario y la contraseña del usuario con el que se está estableciendo la conexión.

Aplicaciones Java

Utilice `getDB2Connection` y `reuseDB2Connection`.

Caso de ejemplo para la implementación de conexiones fiables federadas

Los sistemas federados son exclusivos; por tanto, no existen instrucciones detalladas para la implementación federada de contextos fiables. Utilice estos casos de ejemplo para obtener información completa sobre las conexiones fiables federadas y, a continuación, planifique e implementa su solución.

Cada caso de ejemplo utiliza el mismo sistema federado con múltiples niveles que incluye usuario, una aplicación que se ejecuta en un servidor de aplicaciones, un servidor federado y un servidor de base de datos DB2 remoto. Los casos de ejemplo ilustran la configuración de los orígenes remotos y de los servidores federados para utilizar las conexiones fiables federadas.

El primer caso de ejemplo, que no necesita correlaciones de usuario, es el más simple de implementar y de mantener. De hecho, es el caso de ejemplo recomendado para utilizar cuando se implementan las conexiones fiables en un sistema nuevo.

El segundo caso de ejemplo ilustra la implementación de conexiones fiables en un sistema que incluya correlaciones de usuario. El tercer caso de ejemplo ilustra el aprovechamiento de las conexiones fiables para eliminar correlaciones de usuario. Tenga en cuenta que estos casos de ejemplo son más complejos que el primer caso de ejemplo y requieren un mayor esfuerzo a la hora de ser configurados y mantenidos.

Caso de ejemplo: conexiones fiables federadas de extremo a extremo, sin correlaciones de usuario

Las correlaciones de usuario necesitan un mantenimiento considerable. Este caso de ejemplo ilustra cómo configurar las conexiones fiables federadas de manera que el sistema federado no necesite correlaciones de usuario.

Requisitos de ID y contraseña de usuario

Para configurar los contextos fiables federados sin utilizar correlaciones de usuario, los ID y las contraseñas de usuario deben cumplir los requisitos siguientes:

- El ID y la contraseña de usuario para el originador de la conexión deben estar disponibles para el servidor federado y el ID y la contraseña de usuario del

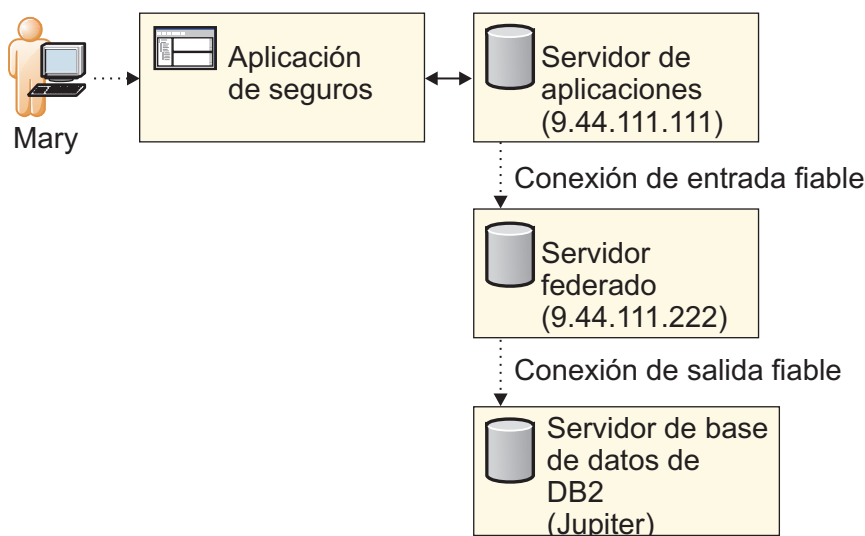
originador deben ser iguales en el servidor federado y en el origen de datos. Las credenciales se identifican en una sentencia CONNECT que envía el originador de la conexión o en una llamada API que realiza la aplicación. Si la sentencia CONNECT se utiliza, se creará una conexión fiable implícita y la conexión no podrá reutilizarse. Si la aplicación realiza una llamada API y solicita explícitamente una conexión fiable, se creará una conexión fiable explícita y se reutilizará la conexión.

- Si el contexto fiable remoto permite la reutilización de la conexión sin autenticación, los usuarios que reutilizan la conexión deben tener el mismo ID en el servidor federado y en el origen de datos remotos.
- Si el contexto fiable remoto permite la reutilización de la conexión con autenticación, los usuarios que reutilizan la conexión deben tener el mismo nombre de usuario y contraseña en el servidor federado y en el origen de datos remotos.

Caso de ejemplo

A continuación se muestra un simple gráfico que ilustra un sistema federado con múltiples niveles típico configurado para utilizar conexiones fiables federadas de extremo a extremo. Aunque el caso de ejemplo incluye un servidor de aplicaciones, cualquier cliente de base de datos puede establecer una conexión fiable.

Originador de la conexión: BOSS



Este caso de ejemplo tiene dos usuarios, ninguno de los cuales necesita correlación de usuario:

- BOSS tiene el mismo ID de usuario y contraseña en el servidor federado y en el origen de datos. Por tanto, BOSS no necesita una correlación de usuario.
- Mary utiliza el mismo ID de usuario en el sistema federado y en el origen de datos y el contexto federado le permite la conexión sin autenticación. Por tanto, no necesita una correlación de usuario.

Este caso de ejemplo tiene tres servidores:

- El servidor de aplicaciones, que aloja la aplicación de seguridad y tiene la dirección IP 9.44.111.111.
- El servidor de federación, que tiene la dirección IP 9.44.111.222.

- El servidor de base de datos DB2 remoto, catalogado en el servidor federado como JUPITER.

Los pasos siguientes describen la configuración de este caso de ejemplo.

Nota: En los mandatos, los nombres de objeto que son variables se muestran en cursiva. Cuando se implementan contextos fiables, especifique nombres de variable que se aplican a la configuración específica del sistema.

1. En el servidor de base de datos DB2 remoto, cree el objeto de contexto fiable:

```
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
```

Este contexto fiable especifica que *BOSS* es el originador de la conexión fiable y que la solicitud para la conexión debe provenir de la dirección IP *9.44.111.222*, que identifica al servidor federado. El contexto fiable especifica *WITH USE FOR PUBLIC WITHOUT AUTHENTICATION*. Después de establecer la conexión fiable, cualquier usuario válido de origen de datos remotos puede reutilizar la conexión proporcionando sólo un ID de usuario.

2. En el servidor federado, cree el objeto de contexto fiable:

```
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
```

Este contexto fiable especifica que *BOSS* es el originador de la conexión fiable y que la solicitud para la conexión debe provenir de la dirección IP *9.44.111.111*, que identifica al servidor de aplicaciones. Después de establecer la conexión fiable, cualquier usuario válido del servidor federado puede reutilizar la conexión proporcionando sólo un ID de usuario.

3. En el servidor federado, cree la definición de servidor siguiente:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb', ...);
```

Esta definición de servidor contiene información que el servidor federado necesita para conectarse a la base de datos DB2 remota llamada *remotedb*.

Caso de ejemplo, paso a paso

A continuación se muestra una descripción detallada sobre cómo se realizan las conexiones fiables y cómo se cambian los usuarios en este caso de ejemplo. El caso de ejemplo incluye los comentarios que describen cómo la aplicación cumple estas tareas.

1. El servidor de aplicaciones solicita una conexión de entrada fiable para *BOSS*.
2. *BOSS* realiza una tarea y el servidor federado establece una conexión fiable de salida explícita para *BOSS*. El ID de usuario *BOSS* se propaga desde el servidor de aplicaciones a través del servidor federado hasta el servidor de base de datos DB2, donde se pueden auditar las acciones que realiza *BOSS*.
3. Mary inicia sesión en la aplicación de seguridad, ubicada en su ordenador. El servidor de aplicaciones cambia la conexión de entrada al servidor federado de *BOSS* a Mary.
4. Mary realiza una tarea dentro de la aplicación.

5. El servidor federado cambia la conexión de salida de BOSS a Mary y su ID se propaga a través del servidor federado al servidor DB2, donde se pueden auditar las acciones que realiza Mary.

Código de muestra para casos de ejemplo de conexión fiable federada de extremo a extremo

Este código de muestra ilustra la utilización de las API en una aplicación que utiliza conexiones fiables federadas de extremo a extremo.

Una aplicación debe utilizar API para solicitar una conexión de entrada fiable explícita y para cambiar el usuario en la conexión. Este código de muestra ilustra las partes de la aplicación que realizan estas tareas. La aplicación es la misma para ambos casos de ejemplo de contexto fiable de extremo a extremo.

Este extracto de aplicación utiliza API de interfaz de línea de mandatos. Las API también están disponibles para aplicaciones que utilizan Java o XA ODBC/CLI.

```
//Establezca el atributo de conexión fiable.
SQLSetConnectAttr(h1, SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER);

//Establezca una conexión de entrada fiable para BOSS.
SQLConnect(h1, "testdb", SQL_NTS, "BOSS", SQL_NTS, "*****", SQL_NTS);

//Establezca una conexión de salida fiable para BOSS.
Realice algunas acciones bajo el ID de usuario BOSS.
SQLExecDirect(hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS);
...
//Confirme el trabajo.
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);

//En el límite de la transacción, cambie el ID de usuario de entrada en la conexión
fiable a Mary.
SQLSetConnectAttr(h1, SQL_ATTR_TRUSTED_CONTEXT_USERID, "Mary", SQL_IS_POINTER);

//Cambie el ID de usuario de salida en la conexión fiable a Mary. Realice algunas acciones bajo el ID
SQLExecDirect(*hstmt, (unsigned char*)"INSERT INTO PATENTS_NN VALUES...", SQL_NTS)

...
//Confirme el trabajo.
SQLEndTran(SQL_HANDLE_DBC, h1, SQL_COMMIT);

//Desconéctese de la base de datos.
SQLDisconnect(h1);
```

Caso de ejemplo: conexiones fiables federadas de extremo a extremo, con correlaciones de usuario.

Las correlaciones de usuario son necesarias cuando los usuarios no tienen el mismo ID de usuario y contraseña en el servidor federado y en el origen de datos remotos. Para este caso de ejemplo, se deben crear correlaciones de usuario federado y se deben configurar contextos fiables.

Requisitos de correlación de usuario

El servidor federado recibe solicitudes de conexión de entrada y realiza solicitudes de conexión de salida a un origen de datos remoto. Si los usuarios tienen el mismo ID de usuario y contraseña en el servidor federado y en el servidor de base de datos DB2 remoto, las correlaciones de usuario no son necesarias. Sin embargo, si las credenciales de usuario no coinciden, es necesaria una correlación de usuario. Las correlaciones de usuario correlacionan el ID de usuario del usuario de servidor

federado al ID de usuario del usuario, y la contraseña de usuario si se ha especificado, de servidor de base de datos remoto.

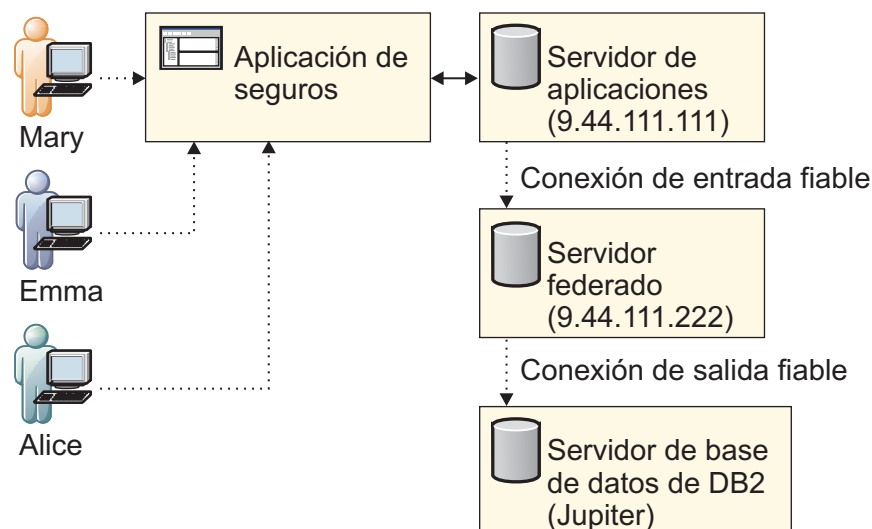
En un sistema federado que utiliza contextos fiables de extremo a extremo, los usuarios cuyo nombre y contraseña de servidor federado y de servidor de base de datos DB2 remoto no coinciden necesitan una *correlación de usuario fiable*. Una correlación de usuario fiable especifica que el usuario tiene permiso para utilizar un contexto fiable. Para crear una correlación de usuario fiable o alterar una correlación de usuario existente, establezca la opción de correlación de usuario `USE_TRUSTED_CONTEXT` en 'Y'.

La creación y alteración de correlaciones de usuario fiable está altamente controlada. Sólo los usuarios con autoridad `SECADM` pueden crear o descartar una correlación de usuario fiable, o bien alterar una correlación de usuario existente para añadir, establecer o descartar la opción de correlación de usuario `USE_TRUSTED_CONTEXT`. Un usuario con correlación de usuario fiable sólo puede alterar la opción `REMOTE_PASSWORD` de su propia correlación de usuario.

Caso de ejemplo

A continuación se muestra un simple gráfico que ilustra un sistema federado con múltiples niveles típico que utiliza correlaciones de usuario. Aunque el caso de ejemplo incluye un servidor de aplicaciones, cualquier cliente de base de datos puede establecer una conexión fiable.

Originador de la conexión: BOSS



Este caso de ejemplo tiene cuatro usuarios:

- BOSS, el originador de la conexión, tiene el mismo ID de usuario y contraseña en el servidor federado y en el servidor de base de datos DB2 remoto. Por tanto, BOSS no tiene ninguna correlación de usuario.
- Mary no tiene ninguna correlación de usuario. Utiliza el mismo ID de usuario tanto en el servidor federado como en el servidor de base de datos DB2. Puesto que el contexto fiable especifica que `PUBLIC` puede reutilizar la conexión sin autenticación, no es necesaria la contraseña de Mary.
- Alice tiene una correlación de usuario que especifica la opción `REMOTE_AUTHID`. Alice tiene un ID de usuario distinto en el servidor

federado y en el servidor de base de datos DB2 remoto. Puesto que el contexto fiable especifica que cualquiera puede reutilizar la conexión sin autenticación, no es necesaria la contraseña de Alice.

- En el servidor federado, Emma utiliza el usuario ID EMMA. Este ID de usuario se correlaciona con el ID de usuario ID EGREENE y la contraseña MYPASS en el servidor de base de datos DB2. Puesto que el contexto federado especifica que Emma se debe autenticar, Emma tiene una correlación de usuario fiable que especifica tanto la opción REMOTE_AUTHID como la opción REMOTE_PASSWORD.

Este caso de ejemplo tiene tres servidores:

- El servidor de aplicaciones, que aloja la aplicación de seguridad y tiene la dirección IP 9.44.111.111.
- El servidor de federación, que tiene la dirección IP 9.44.111.222.
- El servidor de base de datos DB2 remoto, catalogado en el servidor federado como JUPITER.

Para configurar este caso de ejemplo, SECADM completa los pasos siguientes:

1. En el servidor de base de datos DB2 remoto, cree el objeto de contexto fiable:

```
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
```

Este contexto fiable especifica que BOSS es el originador de la conexión fiable y que la solicitud para la conexión debe provenir del servidor federado con la dirección IP 9.44.111.222. Después de establecer la conexión fiable, cualquier usuario de base de datos válido puede reutilizar la conexión proporcionando sólo un ID de usuario.

2. En el servidor federado, cree el objeto de contexto fiable:

```
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
```

Este contexto fiable especifica que BOSS es el originador de la conexión fiable y que la solicitud para la conexión debe provenir de la dirección IP 9.44.111.111, que identifica al servidor de aplicaciones. Después de establecer la conexión fiable, Emma puede reutilizar la conexión, pero se debe autenticar. Cualquier usuario de base de datos válido debe proporcionar sólo un ID de usuario para reutilizar la conexión.

3. En el servidor federado, cree la definición de servidor siguiente:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda
OPTIONS(DBNAME 'remotedb', ...);
```

Esta definición de servidor contiene información que el servidor federado necesita para conectarse a la base de datos DB2 remota llamada remotedb.

4. En el servidor federado, cree la correlación de usuario fiable siguiente para Alice:

```
CREATE MAPPING FOR USER ALICE
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'AJACKSON', USE_TRUSTED_CONTEXT 'Y');
```

Esta correlación de usuario especifica que una conexión fiable puede ser reutilizada por ALICE, que se correlaciona con el usuario ID AJACKSON en el servidor de base de datos DB2 remoto.

5. En el servidor federado, cree la correlación de usuario fiable para Emma:

```
CREATE MAPPING FOR USER EMMA
SERVER JUPITER
OPTIONS
(REMOTE_AUTHID 'EGREENE', REMOTE_PASSWORD 'MYPASS', USE_TRUSTED_CONTEXT 'Y');
```

Esta correlación de usuario especifica que una conexión fiable puede ser reutilizada por EMMA, que se correlaciona con el usuario ID EGREENE y con la contraseña MYPASS en el servidor de base de datos DB2 remoto.

Caso de ejemplo, paso a paso

1. El servidor de aplicaciones solicita una conexión de entrada fiable para BOSS.
2. BOSS realiza una tarea y el usuario ID BOSS se propaga a través del servidor federado al servidor de base de datos DB2, donde se pueden auditar las acciones que realiza BOSS.
3. Emma inicia sesión en la aplicación de seguridad, alojada en el servidor de aplicaciones. El servidor de aplicaciones solicita que la conexión de entrada federada cambie de BOSS a Emma, después de la autenticación de Emma.
4. Emma realiza una tarea dentro de la aplicación.
5. El servidor federado cambia a la conexión de salida federada de BOSS a Emma y el ID de usuario y la contraseña correlacionados se propagan a través del servidor federado al servidor DB2, donde se pueden auditar las acciones que realiza EGREENE (ID de usuario remoto de Emma).
6. Alice inicia sesión en la aplicación de seguridad. El servidor de aplicaciones solicita que la conexión de entrada federada cambie de Emma a Alice, sin la autenticación de Alice.
7. Alice realiza una tarea dentro de la aplicación.
8. El servidor federado cambia a la conexión de salida federada de Emma a AJACKSON (ID de usuario correlacionado de Alice) y su ID de usuario se propaga a través del servidor federado al servidor de base de datos DB2, donde se pueden auditar las acciones que realiza AJACKSON.
9. Mary inicia sesión en la aplicación de seguridad. Mary no necesita autenticarse; por tanto, el servidor de aplicaciones cambia la conexión fiable de entrada federada a Mary, sin proporcionar una contraseña.
10. Mary realiza una tarea dentro de la aplicación.
11. El servidor federado cambia la conexión de salida federada de AJACKSON a Mary y el ID de usuario de Mary se propaga a través del servidor federado al servidor de base de datos DB2, donde se pueden auditar las acciones que realiza Mary.

Caso de ejemplo: conexiones fiables federadas de salida

Una conexión fiable federada de salida establece un entorno fiable entre el servidor federado y el servidor de base de datos DB2 remoto. Puesto que esta conexión no necesita autenticación, se elimina la necesidad de almacenar y conservar las contraseñas de usuario.

Para algunas configuraciones, el servidor federado debe alojar conexiones de entrada no fiables. Una conexión no es fiable cuando se cumple una de las sentencias siguientes:

- Los atributos de la solicitud de conexión de entrada no coinciden con los atributos de ningún objeto de contexto fiable del servidor federado.
- El servidor de federación no especifica un contexto fiable para el servidor desde el cual proviene la solicitud de conexión. Todos los usuarios obtienen conexiones de entrada no fiables.

Contexto fiable, definición de servidor y requisitos de correlación de usuario

Cuando se cree el contexto fiable en el servidor de origen de datos DB2 remoto, debe especificar `WITH USE FOR PUBLIC WITHOUT AUTHENTICATION`. Entonces, todos los usuarios que utilicen el mismo ID de usuario en el servidor federado y en el servidor de base de datos DB2 podrán utilizar el contexto fiable de salida sin autenticación.

Para crear una conexión fiable de salida federada, especifique la opción `FED_PROXY_USER` en la definición de servidor para el servidor de origen de datos DB2. Esta opción identifica el ID de autorización del usuario que origina la conexión fiable de salida. Cree también una correlación de usuario para el usuario de proxy federado. Esta correlación de usuario debe especificar las opciones `REMOTE_AUTHID` y `REMOTE_PASSWORD` porque el contexto fiable de los orígenes de datos necesita la autenticación del originador de la conexión.

Sólo un usuario que tenga autoridad `SECADM` puede crear y alterar una definición de servidor que tenga la opción `FED_PROXY_USER` definida. La sentencia `SET SERVER OPTION` no es válida para la opción de servidor `FED_PROXY_USER`.

Es posible que se produzcan situaciones en las que desee configurar distintos conjuntos de usuarios para conectarse a través de distintos usuarios de proxy. Por ejemplo, si desea configurar distintos roles para distintos usuarios. Para facilitar esta tarea, en cada correlación de usuario que no utilice el usuario de proxy federado predeterminado, especifique la opción `FED_PROXY_USER` y establezca la opción `USE_TRUSTED_CONTEXT` en 'Y'. Cuando la opción `FED_PROXY_USER` esté especificada en la definición de servidor y en la correlación de usuario, el valor de la correlación de usuario alterará temporalmente el valor de la definición de usuario.

Sólo un usuario que tenga autoridad `SECADM` puede añadir, descartar o establecer la opción `FED_PROXY_USER` y crear o descartar una correlación de usuario que incluya la opción `FED_PROXY_USER`.

Importante:

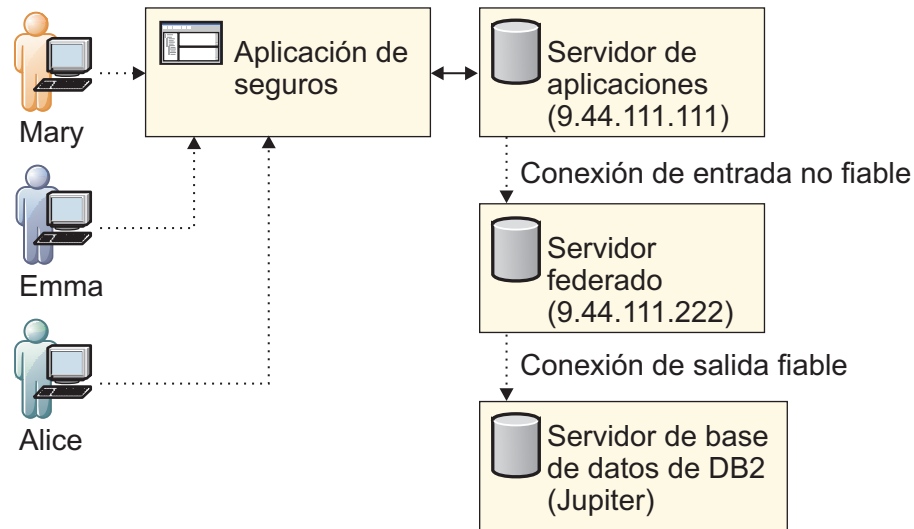
Sólo existen dos situaciones en las que el usuario de una conexión de salida fiable necesite una correlación de usuario: cuando el usuario utiliza un ID de usuario diferente en el servidor federado y en el servidor de base de datos y cuando el usuario se debe conectar a través de un usuario de proxy federado que no es el predeterminado.

Caso de ejemplo

La imagen siguiente ilustra un caso de ejemplo en el que se necesitan conexiones fiables de salida federada. Aunque el caso de ejemplo incluye un servidor de

aplicaciones, cualquier cliente de base de datos puede establecer una conexión fiable.

Usuarios proxy federados: BOSS y ADM



Este caso de ejemplo tiene cinco usuarios:

- BOSS, el usuario de proxy federado predeterminado, tiene una correlación de usuario.
- ADM, el usuario de proxy federado, tiene una correlación de usuario.
- Mary, que utiliza la aplicación de seguridad, no tiene ninguna correlación de usuario.
- Emma y Alice, que utilizan la aplicación de seguridad, tienen correlaciones de usuario.

Este caso de ejemplo tiene tres servidores:

- El servidor de aplicaciones, que aloja la aplicación de seguridad y tiene la dirección IP 9.44.111.111. Este caso de ejemplo muestra un servidor de aplicaciones, pero se puede utilizar cualquier cliente.
- El servidor de federación, que tiene la dirección IP 9.44.111.222.
- El servidor de base de datos DB2 remoto, catalogado en el servidor federado como JUPITER.

Estos pasos describen la configuración:

Nota: En los mandatos, los nombres de objeto que son variables se muestran en cursiva. Cuando se implementan contextos fiables, especifique nombres de variable que se aplican a la configuración específica del sistema.

1. En el servidor de base de datos DB2 remoto, cree el objeto de contexto fiable:

```
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
```

Este contexto fiable especifica que BOSS es el originador de la conexión y que la solicitud para la conexión debe provenir del servidor federado con la

dirección IP 9.44.111.222. Después de establecer la conexión de salida fiable, cualquier usuario puede reutilizar la conexión sin autenticarse.

```
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
```

Este contexto fiable especifica que ADM es el originador de la conexión y que la solicitud para la conexión debe provenir del servidor federado con la dirección IP 9.44.111.222. Después de establecer la conexión de salida fiable, Alice puede reutilizar la conexión sin autenticarse y se le otorgará el rol de Gestora dentro del ámbito de la conexión fiable.

2. En el servidor federado, cree la definición de servidor siguiente:

```
CREATE SERVER JUPITER TYPE db2/udb
VERSION 9.5 WRAPPER drda...
OPTIONS(DBNAME 'remotedb',FED_PROXY_USER 'BOSS');
```

Esta definición de servidor contiene información que el servidor federado necesita para conectarse a la base de datos DB2 remota llamada remotedb. La definición especifica que si la conexión de entrada al servidor federado no es fiable, entonces BOSS, el usuario de proxy federado predeterminado, establece la conexión fiable de salida.

3. En el servidor federado, cree correlaciones de usuario para los usuarios de proxy federado:

```
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. En el servidor federado cree las correlaciones de usuario fiable para Emma y 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');
```

La opción USE_TRUSTED_CONTEXT de ambas correlaciones de usuario está establecida en 'Y', de modo que los usuarios pueden utilizar el contexto fiable. Emma y Alice necesitan las opciones adicionales siguientes:

- Emma necesita una correlación de usuario porque no utiliza el mismo ID de usuario en el servidor federado y en el servidor de base de datos DB2. Por tanto, su correlación de usuario especifica la opción REMOTE_AUTHID.
- Alice necesita una correlación de usuario porque el contexto fiable especifica que utiliza ADM como usuario de proxy federado. Por tanto, su correlación de usuario especifica la opción FED_PROXY_USER.

En este caso de ejemplo, Mary no necesita ninguna correlación de usuario. Las correlaciones de usuario de Emma y Alice no almacenan la contraseña remota; por tanto, dichas correlaciones no necesitan actualizaciones constantes para conservar actualizada la contraseña remota.

Caso de ejemplo, paso a paso

Estos pasos describen el funcionamiento de las conexiones de salida fiable en este escenario.

1. La aplicación crea una conexión de entrada no fiable al servidor federado:

```
CONNECT TO FEDSVR USER MARY USING '****'
```

2. La primera solicitud federada que accede al servidor JUPITER crea una conexión de salida para JUPITER:

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```
3. Puesto que JUPITER especifica FED_PROXY_USER=BOSS, y la correlación de usuario de Mary no especifica ningún usuario de proxy federado, se crea la conexión de salida para BOSS y, a continuación, se cambia inmediatamente al usuario de entrada actual que, en este caso, es Mary.
4. La solicitud federada de Mary ha finalizado y se ha registrado en el registro de auditoría bajo el nombre de MARY. A continuación, se restablecerá la conexión:
5. Cree una conexión de entrada no fiable al servidor federado para Emma:

```
CONNECT TO FEDSVR USER EMMA USING '****'
```
6. La primera solicitud federada de la conexión actual que accede a JUPITER crea una conexión de salida federada a JUPITER.

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```
7. La conexión de salida se crea utilizando BOSS y, a continuación, se cambia inmediatamente al usuario de entrada actual, Emma, cuyo ID se correlaciona con EGREENE.
8. La solicitud federada de Emma ha finalizado y se ha registrado en el registro de auditoría bajo el nombre de EGREENE. A continuación, se restablecerá la conexión:
9. Cree una conexión de entrada no fiable al servidor federado para Alice:

```
CONNECT TO FEDSVR USER ALICE USING '****'
```
10. La primera solicitud federada de la conexión actual que accede a JUPITER crea una conexión de salida federada a JUPITER. Puesto que la correlación de usuario fiable de Alice especifica que utiliza el usuario de proxy federado ADM en lugar del usuario de proxy federado predeterminado BOSS, la conexión de salida se crea utilizando ADM y, a continuación, se cambia inmediatamente a Alice, que es el usuario de entrada actual.

```
SELECT * FROM JUPITER_NN01
CONNECT RESET
```
11. La solicitud federada de Alice ha finalizado y se puede registrar en el registro de auditoría. A continuación, se restablecerá la conexión:

Correlaciones de usuario y conexiones fiables federadas

Estas tablas describen cómo se utilizan las correlaciones de usuario, con y sin ID y contraseñas remotas, en conexiones fiables federadas de extremo a extremo y en conexiones fiables federadas de salida.

Estas tablas describen las posibles correlaciones de usuario para BOSS, el originador de la conexión y para Mary, la usuaria de la conexión. En las tablas, el término *conexión no fiable* hace referencia a una conexión normal que no es fiable en la conexión de entrada ni en la de salida.

Algunas celdas de la tabla utilizan la expresión *si está disponible* para describir una contraseña. Una contraseña está disponible para el servidor federado si el ID y la contraseña de usuario es especifican explícitamente como parte de la conexión. Si utiliza llamadas de conexión API para conectarse al servidor federado, la contraseña se pasará explícitamente; por ejemplo, en CLI/ODBC, la contraseña se especifica a través del atributo de conexión

SQL_ATTR_TRUSTED_CONTEXT_PASSWORD. Si utiliza la sentencia CONNECT para conectarse al servidor federado, utilice esta sintaxis para pasar la contraseña explícitamente.

CONNECT TO *nombre_base de datos* USER *ID usuario* USING *contraseña*

Correlaciones para originadores y usuarios de proxy federado

Tabla 23. Correlaciones de usuario para BOSS, originador de la conexión

Correlación de usuario para BOSS	Conexión no fiable	Conexión fiable federada de extremo a extremo (donde BOSS establece la conexión de entrada fiable)	Conexión fiable federada de salida (donde está establecida la opción de servidor o de correlación de usuario FED_PROXY_USER='BOSS')
Sin correlación de usuario	ID de usuario federado y contraseña federada de BOSS (si está disponible) para el origen de datos remotos.	ID de usuario federado y contraseña federada de BOSS (si está disponible) para el origen de datos remotos.	ERROR SQL1101N
Correlación de usuario que especifica sólo un ID de usuario remoto	ID de usuario remoto y contraseña federada de BOSS (si está disponible) para el origen de datos remotos.	ID de usuario remoto y contraseña federada de BOSS (si está disponible) para el origen de datos remotos.	ERROR SQL1101N
Correlación de usuario que especifica un ID de usuario remoto y una contraseña remota	ID de usuario remoto y contraseña remota de BOSS para el origen de datos remotos.	ID de usuario remoto y contraseña remota de BOSS para el origen de datos remotos.	ID de usuario remoto y contraseña remota de BOSS para el origen de datos remotos.

Correlaciones para usuarios que reutilizan la conexión

Tabla 24. Correlaciones de usuario para Mary, la usuario que reutiliza la conexión

Correlación de usuario para Mary	Conexión fiable federada de extremo a extremo (donde BOSS establece la conexión de entrada fiable y la conexión cambia a Mary)	Conexión fiable federada de salida (donde está establecida la opción de servidor o de correlación de usuario FED_PROXY_USER='BOSS')
Sin correlación de usuario	Pase el ID de usuario federado y la contraseña federada de Mary (si está disponible) al origen de datos remotos.	Pase el ID de usuario federado al origen de datos remotos.
Correlación de usuario no fiable que sólo especifica un ID de usuario remoto	Pase el ID de usuario federado y la contraseña federada de Mary (si está disponible) al origen de datos remotos.	Pase el ID de usuario federado al origen de datos remotos.
Correlación de usuario no fiable que especifica un ID de usuario remoto y una contraseña remota	Pase el ID de usuario federado y la contraseña federada de Mary (si está disponible) al origen de datos remotos.	Pase el ID de usuario federado al origen de datos remotos.

Tabla 24. Correlaciones de usuario para Mary, la usuario que reutiliza la conexión (continuación)

Correlación de usuario para Mary	Conexión fiable federada de extremo a extremo (donde BOSS establece la conexión de entrada fiable y la conexión cambia a Mary)	Conexión fiable federada de salida (donde está establecida la opción de servidor o de correlación de usuario FED_PROXY_USER='BOSS')
Correlación de usuario fiable que sólo especifica un ID de usuario remoto	Pase el ID de usuario remoto de Mary al origen de datos remotos.	Pase el ID de usuario remoto de Mary al origen de datos remotos.
Correlación de usuario fiable que especifica un ID de usuario remoto y una contraseña remota	Pase el ID de usuario remoto de Mary y la contraseña remota al origen de datos remotos.	Pase el ID de usuario remoto de Mary y la contraseña remota al origen de datos remotos.

Capítulo 32. Control de acceso basado en etiquetas (LBAC) y sistemas federados

Asegúrese de que sólo los usuarios que tengan la autoridad adecuada puedan ver los datos de la tabla.

Con el control de acceso basado en etiquetas, podrá aplicar políticas de seguridad a las filas y columnas de una tabla. Las políticas de seguridad especifican las credenciales que se otorgan a cada ID de usuario y de sesión. Por ejemplo, la tabla Precios tiene las columnas Al por mayor, Al por menor y Venta. Si la usuario Alice tiene permiso para acceder a las columnas Al por menor y Venta, entonces las solicitudes `SELECT RETAIL, SALE FROM PRICES` serán satisfactorias. Sin embargo, la solicitud `SELECT * WHOLESale` fallará.

Cuando se crea un apodo para un objeto, el servidor federado detecta automáticamente si el origen de datos utiliza el control de acceso basado en etiquetas. Si se está utilizando el control de acceso basado en etiquetas, el apodo no se almacenará en la memoria caché. Para apodos creados antes de que el control de acceso basado en etiquetas estuviera disponible, utilice la sentencia `ALTER NICKNAME` para permitir o denegar el almacenamiento en la memoria caché. Por ejemplo, si ha creado un apodo en un objeto de origen de datos antes de que el control de acceso basado en etiquetas estuviera disponible, puede alterar el apodo para denegar el almacenamiento en la memoria caché.

Cada política de seguridad tiene una etiqueta única que se almacena en la columna Etiqueta de la tabla. Un administrador de base de datos puede ocultar la columna que contiene las etiquetas para impedir que los usuarios conozcan su existencia. Los apodos que tienen columnas de etiquetas ocultas no se almacenan en la memoria caché.

Capítulo 33. Depósitos de correlaciones de usuarios externos

Almacene las correlaciones de usuario en un único repositorio externo que pueda compartirse entre muchos servidores federados, reduciendo así el mantenimiento administrativo que se necesita cuando se guardan los ID de usuario y las contraseñas en cada servidor federado.

Desde el punto de vista del mantenimiento, almacenar las correlaciones de usuario en un repositorio externo es mejor que almacenarlos en el catálogo. Las contraseñas remotas caducan con regularidad. Si almacena las correlaciones de usuario en el catálogo, cuando una contraseña caduca se debe actualizar en el origen de datos remotos y en el catálogo. Con un repositorio externo, cuando caduca la contraseña remota, sólo se debe actualizar una vez en el repositorio externo.

Para utilizar un repositorio externo para correlaciones de usuario, se debe crear un plugin de correlación de usuario. El plugin debe utilizar valores de seguridad que coincidan con los valores de seguridad que utiliza el repositorio externo. Utilice la capa de sockets seguros (SSL) para establecer conexiones seguras entre el servidor federado y el repositorio externo. A continuación, cuando cree el archivo de configuración para el plugin de correlación de usuario, especifique que el plugin utiliza SSL. Así mismo, restrinja el acceso al código fuente del plugin para que la información esté segura.

Si la auditoría está activada, cada vez que el servidor federado utilice el plugin de correlación de usuario, se creará un registro de auditoría VALIDATE. Para capturar registros VALIDATE, configure el recurso db2audit.

Los plugins y las instrucciones completas para crear, probar y desplegar plugins personalizados se proporcionan en:

Plugin de correlación de usuario (lenguaje de programación C)

El plugin C consta de cinco funciones que proporcionan una interfaz para recuperar correlaciones de usuario desde un repositorio externo.

La secuencia de las llamadas a función se muestra a continuación

1. FSUMPluginInit
2. FSUMconnect
3. FSUMfetchUM
4. FSUMdisconnect
5. FSUMPluginTerm

Inicialización – FSUMPluginInit

Inmediatamente después de que el servidor federado cargue la biblioteca de plugins, el servidor llama a la función FSUMPluginInit, que inicializa el plugin y pasa los punteros de las otras funciones al servidor federado. Estas funciones son globales y se deben resolver de forma externa. Si el plugin está grabado en C++,

estas funciones se deben declarar con "C" externo. El servidor federado pasa a los punteros un conjunto de funciones de programa de utilidad, que el plugin puede obtener, según convenga.

El plugin se ha cargado en el proceso de hebras db2fmp. Todas las aplicaciones que utilizan el plugin comparten la biblioteca de plugins, que debe ser a prueba de riesgos. Cada aplicación utiliza una hebra en el proceso db2fmp. Cuando la aplicación haya recuperado y limpiado satisfactoriamente la correlación de usuario, el servidor federado devolverá la hebra a la agrupación de hebras para utilizarla en el futuro. Puesto que el servidor federado puede servir a múltiples aplicaciones simultáneamente, varias hebras que compartan la misma biblioteca de plugins pueden activarse a la vez. Cada hebra tiene un descriptor de conexiones independiente con el repositorio de correlación de usuario. API FSUMPluginInit también permite que las hebras manejen recursos de plugin global, por ejemplo para aumentar la cuenta de referencias al plugin.

Conexión al repositorio – FSUMconnect

El servidor federado llama a la función FSUMconnect para conectarse al repositorio de correlación de usuario. El plugin debe incluir un descriptor que almacene toda la información necesaria para crear la conexión con el repositorio. Por ejemplo, la información debe incluir un descriptor de contexto para un archivo abierto o un descriptor de conexión para un servidor LDAP. En función de como se implemente la seguridad para el repositorio externo, la información también puede incluir un ID de usuario y contraseña. Si se necesitan las credenciales, se debe configurar el modo de gestión de las mismas. Por ejemplo, si se colocan en un archivo de configuración, cuando el plugin intenta conectarse al repositorio externo, el plugin lee las credenciales desde el archivo.

Recuperación de la correlación de usuario – FSUMfetchUM

El plugin llama a la función FSUMfetchUM para recuperar la correlación de usuario desde el repositorio externo y llama a la función de programa de utilidad FSUMaddUMOption para enviar el ID remoto y la contraseña remota al servidor federado. En el repositorio, las correlaciones de usuario se identifican por el nombre de instancia de servidor federado, el nombre de base de datos, el nombre de servidor remoto y el ID de autorización local. Así mismo, las correlaciones de usuario deben incluir las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Si una contraseña remota está cifrada, el plugin debe descifrarla antes de enviarla al servidor federado.

Desconexión del repositorio – FSUMdisconnect

El servidor federado llama a la función FSUMdisconnect para desconectarse del repositorio de correlación de usuario. Para desconectarse, esta función elimina la asociación entre la hebra y el repositorio de correlación de usuario. Por ejemplo, esta función puede cerrar un archivo abierto o cerrar una conexión con el servidor LDAP.

Liberación de recursos globales – FSUMPluginTerm

Por último, el servidor federado llama a la función FSUMPluginTerm para liberar los recursos globales que ha asignado la función FSUMPluginInit. Para terminar, esta función elimina la asociación entre la hebra y el plugin.

Plataformas soportadas para el plugin de correlación de usuario (lenguaje de programación C)

Antes de compilar el plugin, confirme la utilización de una plataforma soportada.

La tabla siguiente lista las plataformas soportadas para el plugin de correlación de usuario.

Plataforma	Nombre de archivo
AIX, 64 bits	plugin_file_name.a
Linux AMD 64, Power PC, 64, Power PC 390	plugin_file_name.so
Microsoft Windows, 32 bits y 64 bits	plugin_file_name.dll

Restricciones en el desarrollo de un plugin de correlación de usuario (lenguaje de programación C)

Cuando desarrolle plugins de correlación de usuario en C, tenga en cuenta estas restricciones.

C-linkage

La biblioteca de plugins debe estar enlazada con C-linkage. Los archivos de cabecera que proporcionan los prototipos, las estructuras de datos necesarias para implementar el plugin y las definiciones de código de error sólo se proporcionan para C/C++. Las funciones que se resolverán durante el tiempo de carga se deben declarar con "C" externo si la biblioteca de plugins se ha compilado como C++.

El tiempo de ejecución de lenguaje común .NET no se soporta.

El tiempo de ejecución de lenguaje común .NET no se soporta para compilar y enlazar el código fuente para la biblioteca de plugins.

Manejadores de señal

La biblioteca de plugins no debe instalar manejadores de señal ni cambiar la máscara de señal porque interferiría en la información y recuperación de errores. La biblioteca de plugins nunca debe emitir excepciones C++.

A prueba de riesgos

La biblioteca de plugins debe ser a prueba de riesgos y reentrante. Únicamente la inicialización del plugin puede no ser reentrante. La función de inicialización del plugin puede ser llamada varias veces desde distintas hebras, en cuyo caso, el plugin limpiará todos los recursos utilizados y se reinicializará.

Alteración temporal de la biblioteca C estándar y de las llamadas de sistema operativo

La biblioteca de plugins no debe alterar temporalmente la biblioteca C estándar ni las llamadas de sistema operativo.

Aplicaciones de 32 bits y de 64 bits

Un servidor federado de 32 bits debe utilizar un plugin de 32 bits. Un servidor federado de 64 bits debe utilizar un plugin de 64 bits. En una instancia híbrida, donde el cliente es de 32 bits y el servidor de 64 bits, el plugin debe ser de 64 bits.

Series de texto

No se garantiza que las series de entrada sean de terminación nula y no es necesario que las series de salida lo sean. En cambio, se proporcionan

longitudes enteras para todas las series de entrada y se proporcionan punteros a los enteros para que se devuelvan longitudes.

Archivo de cabecera (fsumplugin.h) para el plugin de correlación de usuario (lenguaje de programación C)

El archivo de cabecera, fsumplugin.h, contiene estructuras de datos, funciones y códigos de error.

El plugin de correlación de usuario debe incluir este archivo de cabecera que se encuentra en el directorio sqllib/include.

```
/* Definición de nombres de opción de correlación de usuario.*/
#define FSUM_REMOTE_AUTHID_OPTION "REMOTE_AUTHID"
#define FSUM_REMOTE_PASSWORD_OPTION "REMOTE_PASSWORD"

/* Definición de tipos de valor de opción.*/
#define FSUM_OPTION_VALUE_BINARY_TYPE 1
#define FSUM_OPTION_VALUE_STRING_TYPE 2

/* Estructura de datos para describir una opción de usuario.*/

typedef struct _FSUMOption
{
    const char* optionName;
    size_t optionNameLen;
    char* optionValue;
    size_t optionValueLen;
    size_t optionValueType;
    struct _FSUMOption* nextOption;
} FSUMOption;

/* Estructura de datos para describir una entrada de correlación de usuario. Una entrada de correlación
Estas opciones se encuentran en una lista enlazada. Esta estructura de datos conserva el puntero a la siguiente entrada.*/

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;

/* Las funciones para implementar junto con la función FSUMPluginInit,
que no está en la estructura 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;
```



```

/* El servidor federado proporciona estos programas de utilidad como funciones de devolución de llamada al plugin. */

typedef SQL_API_RC (SQL_API_FN FSUMallocateFP)
(size_t a_blkSize, void** a_pblkPtr);

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);

/* Estructura para conservar las funciones de programa de utilidad que proporciona el servidor federado */

typedef struct _FSUMPluginUtilities
{
    FSUMallocateFP    *allocate;
    FSUMdeallocateFP *deallocate;
    FSUMloadFP       *load;
    FSUMgetFunctionFP *getFunction;
    FSUMunloadFP     *unload;
    FSUMlogErrorMsgFP *logErrorMsg;
    FSUMaddUMOptionFP *addUMOption;
} FSUMPluginUtilities;

/* Tipo de punto de entrada de plugin de correlación de usuario.*/
typedef SQL_API_RC (SQL_API_FN *FSUMPluginInitType)
(sqlint32, FSUMPluginAPIs*, FSUMPluginUtilities*);

/* Tipo de punto de entrada de interfaz C de plugin de correlación de usuario.*/
typedef SQL_API_RC (*fsum_plugin_hook_type)
(const char*, FSUMPluginInitType*, FSUMPluginUtilities*);

/* Definición de códigos de retorno para funciones de utilidad */
#define FSUM_PLUGIN_UTIL_OK 0
#define FSUM_PLUGIN_UTIL_FAILED -1

/* Definición de C externo. */
#define FSUM_PLUGIN_EXT_C extern "C"

/* Gravedad de error que utiliza la función logErrorMsg.*/
#define FSUM_LOG_NONE      0 /* Sin registro */
#define FSUM_LOG_CRITICAL 1 /* Error grave encontrado */
#define FSUM_LOG_ERROR     2 /* Error encontrado */
#define FSUM_LOG_WARNING   3 /* Aviso */
#define FSUM_LOG_INFO      4 /* Informativo */

/* Códigos de error que devuelve el retorno.*/
#define FSUM_PLUGIN_OK          0
#define FSUM_INITIALIZE_ERROR  1

```

```

#define FSUM_PLUGIN_VERSION_ERROR    2
#define FSUM_CONNECTION_ERROR       3
#define FSUM_LOOKUP_ERROR            4
#define FSUM_DECRYPTON_ERROR          5
#define FSUM_DISCONNECT_ERROR        6
#define FSUM_INVALID_PARAMETER_ERROR 7
#define FSUM_UNAUTHORIZED_CALLER     8
#define FSUM_AUTHENTICATION_ERROR    9
#define FSUM_TERMINATION_ERROR       10

/* Longitud máxima del nombre.*/
#define FSUM_MAX_NAME_LEN            128

/* Longitud máxima de una vía de acceso de archivo. */
#define FSUM_MAX_PATH_LEN            256

/* Longitud máxima de un valor de opción. */
#define FSUM_MAX_OPTION_VALUE_LEN    (2048+1)

/* Longitud máxima de un mensaje de error.*/
#define FSUM_MAX_ERROR_MSG_SIZE      2048

```

Función FSUMPluginInit (lenguaje de programación C)

FSUMPluginInit es la función de inicialización para el plugin de correlación de usuario.

Esta función realiza las tareas siguientes:

- Pasa los punteros para las otras cuatro funciones del servidor federado.
- Obtiene las funciones de programa de utilidad que pasa el servidor federado.
- Configura los recursos globales en el nivel del plugin.

Sintaxis

```

SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
FSUMPluginAPIs* a_pluginAPIs,
FSUMPluginUtilities* a_pluginUtils);

```

Entradas

sqlint32 a_version

El número de versión de la interfaz del plugin de correlación de usuario.

FSUMPluginUtilities *a_pluginUtils

El puntero de la estructura que contiene todas las funciones de programa de utilidad. El plugin obtiene un puntero para cada función de programa de utilidad, según convenga.

Salidas

FSUMPluginAPIs *a_pluginAPIs

El plugin utiliza esta estructura para pasar los punteros de función para FSUMconnect, FSUMfetchUM, FSUMdisconnect y FSUMPluginTerm al servidor federado.

Función FSUMconnect (lenguaje de programación C)

Para conectarse al repositorio de usuario externo, el servidor federado llama a la función FSUMconnect.

Sintaxis

```
SQL_API_RC SQL_API_FN FSUMconnect  
(void** a_FSUMRepository,  
const char* a_cfgFilePath)
```

Entradas

const char* a_cfgFilePath

Vía de acceso completa a la biblioteca del plugin. Si el archivo de configuración se sitúa en el mismo directorio que la biblioteca del plugin, éste puede utilizar la información de la vía de acceso para ubicar el archivo de configuración.

Salidas

void a_FSUMRepository**

El puntero del descriptor de conexión se convierte en absoluto antes de enviarse al servidor federado. En las siguientes llamadas de función API, el servidor federado pasa el puntero al plugin y el plugin debe convertir el puntero de nuevo a la estructura real del descriptor de conexión.

Función FSUMfetchUM (lenguaje de programación C)

El servidor federado llama a la función FSUMfetchUM para recuperar las opciones de correlación de usuario desde un repositorio externo.

Cada entrada de correlación de usuario se identifica por el nombre de la instancia federada (fsInstanceName), por el nombre de base de datos (fsDatabaseName), por el nombre del servidor remoto (fsServerName) y por el ID de autorización de usuario (fsAuthID). La correlación de usuario también puede incluir las opciones REMOTE_AUTHID y REMOTE_PASSWORD, que recuperan el plugin.

Sintaxis

```
SQL_API_RC SQL_API_FN FSUMfetchUM (void* a_FSUMRepository,  
FSUMEntry* a_entry);
```

Entradas

void* a_FSUMRepository

El descriptor de conexión del repositorio externo. Este descriptor de contexto debe asignarse a la estructura real definida en el plugin.

FSUMEntry* a_entry

Este parámetro es un parámetro de entrada y de salida simultáneamente.

Como parámetro de entrada, pasa información necesaria para identificar la entrada de correlación de usuario en el repositorio de correlación. Esta información incluye fsInstanceName, fsDatabaseName, fsServerName y fsAuthID.

Como parámetro de salida, se añadirán las opciones de correlación de usuario recuperadas desde el repositorio de correlación de usuario. El plugin debe utilizar la función de programa de utilidad FSUMaddUMOption para añadir la opción a la entrada de correlación de usuario, a_entry. Si la correlación de usuario incluye la opción REMOTE_PASSWORD y la contraseña está cifrada, el plugin debe descifrarla antes de llamar a FSUMaddUMOption. El servidor federado no se responsabiliza de la liberación de almacenamiento para las series que se pasan a la función FSUMaddUMOption.

Función FSUMdisconnect (lenguaje de programación C)

Para desconectarse del repositorio de usuario externo, el servidor federado llama a esta función.

Sintaxis

```
SQL_API_RC SQL_API_FN FSUMdisconnect  
(void* a_FSUMRepository);
```

Entradas

void* a_FSUMRepository

El descriptor de conexión del repositorio externo. Este descriptor de contexto debe asignarse a la estructura real definida en el plugin.

Función FSUMPluginTerm (lenguaje de programación C)

El plugin de correlación de usuario llama a esta función para liberar recursos globales que asigna la función FSUMPluginInit.

Esta función no tiene parámetros.

```
SQL_API_RC SQL_API_FN FSUMPluginTerm ()
```

Desarrollo del plugin de correlación de usuario (lenguaje de programación C)

El plugin que desarrolle debe conectarse al repositorio externo, recuperar las correlaciones de usuario y descifrar las contraseñas remotas. El repositorio que se utiliza determina cómo codificar el plugin.

Importante: Tenga en cuenta que cuando envíe y utilice el plugin, enviará ID y contraseñas de usuario sensibles entre múltiples orígenes. Para proteger esta información, restrinja el acceso al código fuente del plugin y configure el programa de utilidad de auditoría de DB2 para capturar un registro VALIDATE en el archivo de registro de diagnóstico cada vez que el servidor federado utilice el plugin. El archivo de registro de diagnóstico es de gran utilidad para realizar seguimientos de uso pero también para resolver los problemas que se puedan llegar a producir.

Para desarrollar un plugin de correlación de usuario, realice las tareas siguientes:

Declaración de funciones externas para el plugin de correlación de usuario (lenguaje de programación C)

El plugin implementa las funciones. Cuando se llama a la función FSUMPluginInit, los punteros de función se pasan al servidor federado.

La función de inicialización, FSUMPluginInit, es la única función que debe tener exactamente este nombre de función. Para las funciones restantes, se puede utilizar cualquier nombre de función C válido. Las funciones son globales y se deben resolver de forma externa. Si graba el plugin en C++, debe utilizar FSUM_PLUGIN_EXT_C para declarar las funciones.

El plugin de muestra implementa las API siguientes:

- API FSUMconnect en la función myConnect
- FSUMfetchUM en la función myFetchUM
- FSUMdisconnect en la función myDisconnect
- FSUMPluginTerm en la función myPluginTerm

En el plugin de muestra, cuando se llama a la función FSUMPluginInit, los punteros de función de myConnect, myFetchUM, myDisconnect y myPluginTerm se pasan al servidor federado.

El código siguiente está en el archivo 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 ()
```

Declaración de funciones del programa de utilidad para el plugin de correlación de usuario (lenguaje de programación C)

Debe declarar cuatro opciones del programa de utilidad necesarias para el plugin de correlación de usuario.

FSUMlogErrorMsg, FSUMaddUMOption, FSUMallocate y FSUMdeallocate son funciones del programa de utilidad necesarias para el plugin de correlación de usuario. Las funciones del programa de utilidad opcionales pueden ser útiles al desarrollar un plugin de correlación de usuario. Obtenga los punteros de las funciones del programa de utilidad desde la estructura FSUMPluginUtilities en la función FSUMPluginInit.

Funciones del programa de utilidad necesarias

```
FSUMlogErrorMsgFP *FSUMlogErrorMsg=NULL;
FSUMaddUMOptionFP *FSUMaddUMOption=NULL;
FSUMallocateFP *FSUMallocate=NULL;
FSUMdeallocateFP *FSUMdeallocate=NULL;
```

Funciones del programa de utilidad opcionales

```
FSUMgetFunctionFP *FSUMgetFunction=NULL;
FSUMloadFP *FSUMload=NULL;
FSUMunloadFP *FSUMunload=NULL;
```

Establecimiento de punteros de función para el plugin de correlación de usuario (lenguaje de programación C)

Pase los punteros de las funciones necesarias al servidor federado y obtenga los punteros para las funciones de utilidad del servidor federado.

/ La función de inicialización del plugin. No cambie el nombre. */*

```
SQL_API_RC SQL_API_FN FSUMPluginInit
(sqlint32 a_version,
FSUMPluginAPIs* a_pluginAPIs,
FSUMPluginUtilities* a_pluginUtils)
{
```

```
    SQL_API_RC rc = FSUM_PLUGIN_OK ;
```

```
    /* Pase los punteros de función al servidor federado */
```

```
    a_pluginAPIs->FSUMconnect = &myConnect
```

```

a_pluginAPIs->FSUMfetchUM = &myFetchUM
a_pluginAPIs->FSUMdisconnect = &myDisconnect
a_pluginAPIs->FSUMPluginTerm = &myPluginTerm

/* Obtenga los punteros de las funciones de programa de utilidad necesarias */

FSUMallocate = a_pluginUtils->allocate;
FSUMdeallocate = a_pluginUtils->deallocate;
FSUMlogErrorMsg = a_pluginUtils->logErrorMsg;
FSUMaddUMOption = a_pluginUtils->addUMOption;
/*También puede obtener los punteros de las funciones de programa de utilidad opcionales */

/** Si se produce un error, realice las acciones siguientes:
  1. Opcional --- Llame a FSUMlogErrorMsg para registrar el error
  2. Obligatorio --- Devuelva el código de error rc = FSUM_INITIALIZE_ERROR
**/

return rc;
}

```

Error al manejar el plugin de correlación de usuario (lenguaje de programación C)

El plugin de correlación de usuario utiliza la función FSUMlogErrorMsg para registrar todos los errores y mensajes informativos en el archivo db2diag.log.

Si una llamada a función es satisfactoria, el plugin devuelve FSUM_PLUGIN_OK. Si la llamada no es satisfactoria, el plugin devuelve un código de error. La tabla siguiente describe los errores.

Tabla 25. Errores de plugin de correlación de usuario

Número de error	Nombre de constante	Mensaje de error	Funciones que devuelven el error	Código de error
0	FSUM_PLUGIN_OK	El plugin se ha ejecutado satisfactoriamente.	<ul style="list-style-type: none"> FSUMPluginInit FSUMconnect FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL_SUCCESS (Sin error)
1	FSUM_INITIALIZE_ERROR	El plugin no se ha podido inicializar.	<ul style="list-style-type: none"> FUMPluginInit 	SQL20349N , código de razón 1
2	FSUM_PLUGIN_VERSION_ERROR	La versión del plugin es incorrecta.	<ul style="list-style-type: none"> FUMPluginInit 	SQL20349N, código de razón 2
3	FSUM_CONNECTION_ERROR	No se puede conectar con el repositorio externo.	<ul style="list-style-type: none"> FSUMPluginInit 	SQL20349N, código de razón 3
4	FSUM_LOOKUP_ERROR	La búsqueda en el repositorio ha fallado.	<ul style="list-style-type: none"> FSUMfetchUM 	SQL20349N, código de razón 4
5	FSUM_DECRYPTION_ERROR	El descifrado ha fallado.	<ul style="list-style-type: none"> FSUMfetchUM 	SQL20349N, código de razón 5
6	FSUM_DISCONNECT_ERROR	No se pueden desconectar desde el repositorio.	<ul style="list-style-type: none"> FSUMdisconnect 	SQL20349N, código de razón 6
7	FSUM_INVALID_PARAMETER_ERROR	Parámetro no válido.	<ul style="list-style-type: none"> FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL20349N, código de razón 7

Tabla 25. Errores de plugin de correlación de usuario (continuación)

Número de error	Nombre de constante	Mensaje de error	Funciones que devuelven el error	Código de error
8	FSUM_UNAUTHORIZED_CALLER	El interlocutor no está autorizado a llamar al plugin.	<ul style="list-style-type: none"> FSUMPluginInit 	SQL20349N, código de razón 8
9	FSUM_AUTHENTICATION_ERROR	No se puede autenticar con el repositorio.	<ul style="list-style-type: none"> FSUMconnect 	SQL20350N, sin código de razón
10	FSUM_TERMINATION_ERROR	No se ha podido limpiar un recurso en el nivel del plugin.	<ul style="list-style-type: none"> FSUMPluginTerm 	SQL20349N, código de razón 9
No es número del 0 al 10		Se ha producido un error desconocido.	<ul style="list-style-type: none"> FSUMPluginInit FSUMconnect FSUMfetchUM FSUMdisconnect FSUMPluginTerm 	SQL20349N, código de razón 10

El plugin también puede utilizar la función de programa de utilidad FSUMlogErrorMsg para registrar mensajes de registro en el archivo db2diag.log. Para especificar la gravedad de los mensajes, utilice FSUM_LOG_CRITICAL, FSUM_LOG_ERROR, FSUM_LOG_WARNING o FSUM_LOG_INFO.

Cuando utilice el programa de prueba para probar el plugin, los mensajes se imprimirán directamente en la pantalla.

Creación del plugin de correlación de usuario (C)

Utilice el script bldplugin para crear el plugin de correlación de usuario

Para crear el plugin de correlación de usuario:

1. Examine el script bldplugin y actualice la información de la vía de acceso para que coincida con la instalación específica.
2. Emita este mandato para crear el plugin:
`bldplugin nombre_archivo_plugin`
 Para obtener más información, consulte el archivo bldplugin.
3. Configure el repositorio externo para incluir la información de correlación de usuario y reforzar un esquema de cifrado. Si se está creando un plugin de correlación de usuario de muestra, abra el archivo fsumplugin_file.txt y cree una nueva entrada de correlación de usuario para el sistema. El plugin de muestra utiliza un esquema de cifrado simple que invierte los bytes de la contraseña remota.

Prueba del plugin de correlación de usuario (lenguaje de programación C)

Utilice este programa de prueba para cargar y probar el plugin de correlación de usuario de muestra o el plugin de correlación de usuario personalizado antes de desplegarlo en el servidor federado.

El programa de configuración fsumsetup_file (fsumsetup_file.exe en Microsoft Windows) y el programa de prueba fsumlookup (fsumlookup.exe en Windows) están instalados en el mismo directorio que los archivos del plugin de muestra.

El programa `fsumsetup_file` genera un archivo de configuración llamado `fsumplugin_file.cfg`. En este archivo se almacena la vía de acceso completa al archivo de texto que contiene las entradas de correlación de usuario. El programa `fsumlookup` (`fsumlookup.exe` en Microsoft Windows) prueba el plugin.

1. Ejecute el programa `fsumsetup_file` para configurar la vía de acceso completa y el nombre del archivo que almacena las correlaciones de usuario.
2. Ejecute el programa `fsumlookup` para probar el plugin:

```
fsumlookup pluginName fsInstance fsDatabase fsRemoteServer fsAuthid
```

donde:

- *pluginName* es el nombre del plugin.
- *fsInstance* es el nombre de la instancia del servidor federado.
- *fsDatabase* es el nombre de la base de datos federada.
- *fsRemoteServer* es el nombre del servidor de origen de datos remotos, tal como se especifica en la sentencia CREATE SERVER.
- *fsAuthid* es el ID de usuario local.

Si el plugin funciona correctamente, el programa `fsumlookup` imprimirá los parámetros que identifican la entrada de correlación de usuario, así como los nombres de opción y los valores de opción, en la pantalla. A continuación se muestra un ejemplo de la salida.

```
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 se produce un error, el programa de prueba imprimirá un mensaje de error en la pantalla.

Despliegue del plugin de correlación de usuario (lenguaje de programación C)

Puede crear y compilar un plugin de correlación de usuario en cualquier directorio. A continuación, despliegue el plugin y cópielo en el directorio correcto del servidor federado.

Para desplegar el plugin en el servidor federado, copie el plugin y el archivo de configuración en el directorio correcto del servidor federado.

- En UNIX y Linux, *directorio_instalación*/sqllib/function, donde *directorio_instalación* es el directorio de la instancia
- En Windows, %DB2PATH%\function, donde %DB2PATH% es el directorio donde está instalado el sistema de base de datos CB2

Habilitación del acceso al repositorio de correlación de usuario externo

Después de crear un plugin de correlación de usuario, debe establecer las opciones `DB2_UM_PLUGIN` y `DB2_UM_PLUGIN_LANG` para habilitar el acceso para el servidor federado a las correlaciones de usuario almacenadas en el repositorio externo.

Puede establecer las opciones en el derivador o en el nivel de servidor, pero se deben establecer las dos opciones al mismo nivel. `DB2_UM_PLUGIN` especifica el nombre del plugin y `DB2_UM_PLUGIN_LANG` especifica el lenguaje en que se

graba el plugin. Las dos opciones son interdependientes. Por tanto, se debe añadir DB2_UM_PLUGIN antes de añadir DB2_UM_PLUGIN_LANG. Si se añaden ambas opciones a la misma sentencia, el orden no importa. Para descartar las opciones, se debe descartar antes DB2_UM_PLUGIN_LANG que DB2_UM_PLUGIN.

Por ejemplo, las sentencias siguientes utilizan el mismo plugin para crear el derivador w1, alterar el derivador w2, crear el servidor s1 y alterar el servidor s2 para utilizar el plugin de correlación de usuario:

```
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');
```

Actualización del plugin de correlación de usuario (lenguaje de programación C)

Después de modificar un plugin de correlación de usuario, se debe copiar la versión actualizada al servidor federado.

Para actualizar el plugin de correlación de usuario:

1. Copie el plugin actualizado en el directorio sqllib/function.
2. Entre los mandatos siguientes para detener y, a continuación, reiniciar la instancia de servidor federado:

```
db2stop
db2start
```

Plugin de correlación de usuario de muestra (lenguaje de programación C)

Revise el código fuente de muestra para obtener más información sobre el desarrollo de un plugin que proporcione una interfaz C a un repositorio de correlación de usuario externo.

Este plugin de correlación de muestra utiliza un archivo como repositorio externo. Los archivos de origen de muestra se instalan automáticamente en el sistema. La ubicación donde se instalen dependerá del sistema operativo que se utilice:

- En Windows, %DB2PATH%\samples\federated\umplugin\file, donde %DB2PATH% es el directorio donde está instalado el sistema de base de datos DB2
- En Linux y UNIX, *directorio_instalación*/sqllib/samples/federated/umplugin/file, donde *directorio_instalación* es el directorio de la instancia

La muestra incluye los archivos siguientes:

fsumplugin_file.h

El archivo de cabecera que contiene las estructuras de datos y las definiciones de función para el plugin de muestra.

fsumplugin_file.c

El código fuente para el plugin de muestra, que utiliza un archivo como repositorio externo.

fsumplugin_file.txt

Un archivo que contiene entradas de correlación de usuario de muestra en texto sin formato.

fsumlookup

Un programa autónomo que prueba el plugin de muestra fuera del entorno del servidor federado. En Microsoft Windows, el archivo es `fsumlookup.exe`.

fsumsetup_file

Un programa autónomo que establece la vía de acceso completa y el nombre de archivo del archivo que almacena las correlaciones de usuario. En Microsoft Windows, el archivo es `fsumsetup_file.exe`.

bldplugin

El script que compila y enlaza el plugin. En Microsoft Windows, el archivo es `bldplugin.bat`.

README Un archivo que contiene instrucciones para compilar, crear, probar y desplegar el plugin de muestra.

En el archivo `fsumplugin_file.txt`, cada entrada de correlación de usuario ocupa una línea y las contraseñas están cifradas mediante la inversión de los bytes. Se trata de un método de cifrado simple. Si se modifica el plugin de muestra para utilizarlo en un entorno de producción, se debe utilizar un método de cifrado que cumpla los requisitos de seguridad del entorno.

La sintaxis siguiente muestra el formato para una entrada de correlación de usuario de muestra.

Nota: A pesar de que el código se muestra en múltiples líneas, se debe entrar todo el código en una línea de la aplicación. Tenga en cuenta que un punto y coma separa las partes de la correlación de usuario y dos puntos separan los nombres de opción `REMOTE_AUTHID` y `REMOTE_PASSWORD` de sus valores correspondientes.

```
fsInstance;fsDatabase;fsRemoteServer;  
fsAuthid;REMOTE_AUTHID:remote_authID;  
REMOTE_PASSWORD:remote_password
```

Por ejemplo:

```
DB2INST1;TESTDB;MSSQL2K;ALICE;REMOTE_AUTHID:TESTUSR1;REMOTE_PASSWORD:drowssap
```

Plugin de correlación de usuario (lenguaje de programación Java)

La arquitectura del plugin de Java incluye dos clases de interfaz y tres clases de programa de utilidad. Utilícelas para desarrollar un plugin que recupera las correlaciones de usuario de un repositorio externo.

De forma predeterminada, las correlaciones de usuario para un origen de datos se almacenan localmente en cada servidor federado. Un servidor LDAP almacena objetos, como entradas de correlación de usuario, en un árbol de directorio. Los

objetos pueden ser atributos, como contraseñas. Así mismo, el servidor LDAP a menudo almacena información adicional sobre los usuarios, por ejemplo las direcciones de correo electrónico y los números de teléfono.

La imagen siguiente ilustra la arquitectura del plugin de correlación de usuario de muestra:

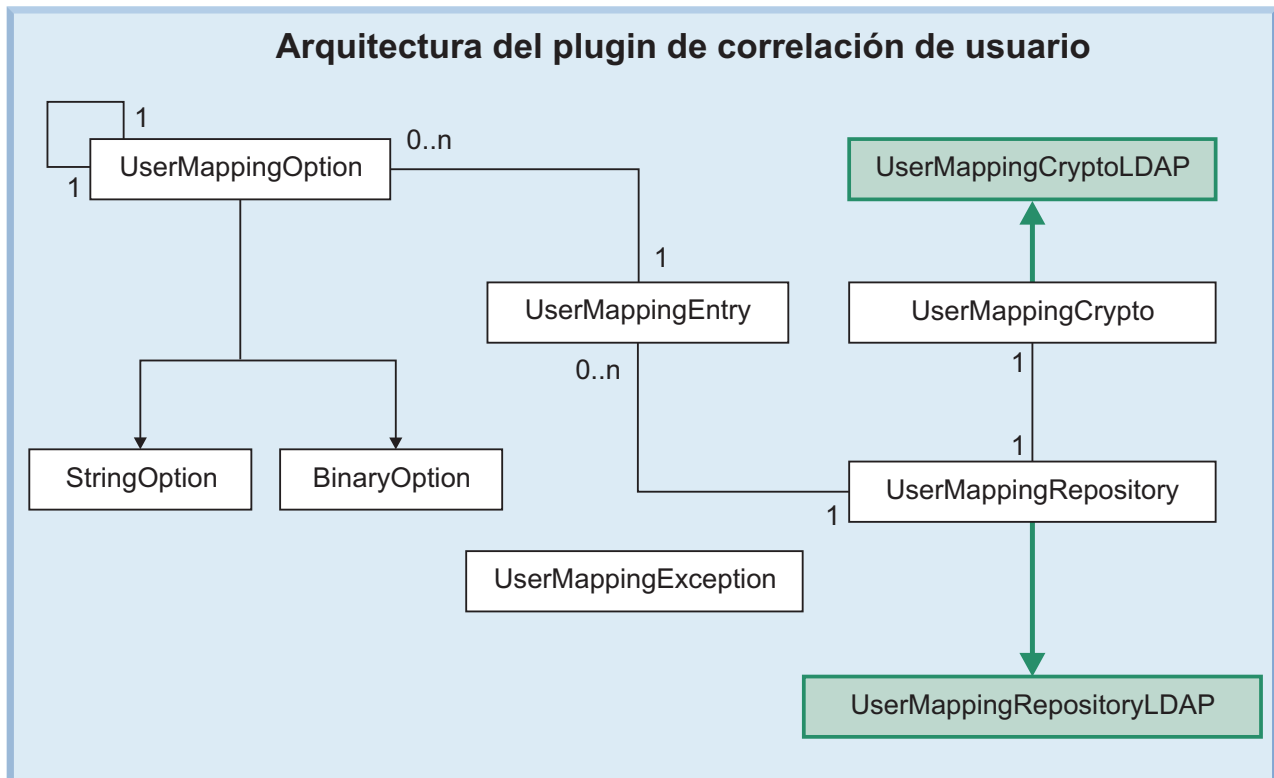


Figura 15. Arquitectura del plugin de correlación de usuario

UserMappingCrypto y UserMappingRepository son clases de interfaz. Para crear un plugin de correlación de usuario que recupere correlaciones de usuario desde el repositorio externo que utiliza la organización, se deben ampliar las clases de interfaz. UserMappingEntry, UserMappingOption y UserMappingException son clases de programa de utilidad. Puede utilizar las clases de programa de utilidad sin necesidad de modificarlas. Algunos métodos y funciones de las clases de la interfaz actúan como funciones de programa de utilidad. Por ejemplo, puede utilizar las funciones getChars() y getBytes(), de la clase UserMappingCrypto, sin necesidad de modificarlas.

En la imagen, el término *0..n* significa que puede haber 0 o más objetos. Por ejemplo, un objeto UserMappingEntry puede tener múltiples objetos UserMappingOption, pero sólo puede haber un objeto UserMappingRepository. Las clases UserMappingCryptoLDAP y UserMappingRepositoryLDAP se amplían desde sus clases padre.

Clases para el plugin de correlación de usuario (lenguaje de programación Java)

La arquitectura del plugin Java incluye dos clases de interfaces y tres clases de programa de utilidad. Utilícelas para desarrollar un plugin que recupera las correlaciones de usuario de un repositorio externo.

Clase `UserMappingRepository` (lenguaje de programación Java)

La clase `UserMappingRepository` es una clase abstracta que no tiene constructor. Para crear el plugin de correlación de usuario, se debe crear una subclase de la clase `UserMappingRepository` o modificar la subclase en el plugin Java de muestra.

La clase `UserMappingRepository` contiene los métodos públicos siguientes: `getVersionNumber()`, `getCrypto()`, `connect()`, `disconnect()`, `fetchUM()`, y `lookupUM()`. Debe crear sus propias subclases de `UserMappingRepository` que contengan estas funciones. En estas funciones, escriba el código que utiliza el plugin para interactuar con el repositorio externo.

Puede ver la implementación de estas funciones en un plugin Java de muestra que recupera correlaciones de usuario de un servidor LDAP. Los archivos se encuentran en el directorio `sql/lib/samples/federated/umplugin/ldap/`. Las funciones de esta clase se utilizan en los archivos `UserMappingRepositoryLDAP.java` y `UserMappingLookupLDAP.java`.

Métodos públicos

`int getVersionNumber()`

Devuelve el número de versión del kit de desarrollo del plugin que utiliza el plugin.

`UserMappingCrypto getCrypto()`

Devuelve el objeto `UserMappingCrypto` asociado con este objeto `UserMappingRepository`.

`abstract void connect()`

Debe implementar su propio método de conexión con el repositorio dentro de esta función.

`abstract void disconnect()`

Debe implementar su propio método de desconexión con el repositorio dentro de esta función.

`abstract void fetchUM(UserMappingEntry um)`

Debe implementar su propio método de recuperación de la correlación de usuario desde el repositorio dentro de esta función. El parámetro `um` contiene la información de consulta detallada utilizada para determinar qué correlación de usuario se debe recuperar.

`UserMappingEntry lookupUM(UserMappingRepository repository, String iiInstanceName, String iiDatabaseName, String iiRemoteServerName, String iiAuthid)`

Esta función se utiliza principalmente para probar el plugin. La función utiliza los parámetros `iiInstanceName`, `iiDatabaseName`, `iiRemoteServerName` y `iiAuthid` como entrada para crear e inicializar la clase `UserMappingEntry`. La función llama a los métodos `connect`, `fetchUM` y `disconnect`.

Clase `UserMappingCrypto` (lenguaje de programación Java)

Si el repositorio externo cifra o codifica contraseñas remotas, debe crear una subclase de la clase `UserMappingCrypto`. El constructor de la subclase creada se

utilizará para construir el objeto criptográfico. Los métodos de clase de criptografía son llamados por otras clases cuando es necesario cifrar, descifrar, codificar o descodificar las contraseñas de correlación de usuario.

La clase `UserMappingCrypto` contiene los métodos públicos siguientes: `encrypt()`, `decrypt()`, `encode()` y `decode()`. En estas funciones, se debe escribir el código para cifrar, descifrar, codificar y descodificar la contraseña remota. Las funciones `getBytes()` y `getChars()` son funciones de utilidad que se heredan y pueden utilizarse sin modificaciones. Los métodos de cifrado, descifrado, codificación y descodificación que se codifican deben coincidir con los métodos de cifrado y descifrado que utiliza el repositorio externo para proteger las contraseñas almacenadas.

Puede ver la implementación de estas funciones en un plugin Java de muestra que recupera correlaciones de usuario de un servidor LDAP. Los archivos se encuentran en el directorio `sql11lib/samples/federated/umplugin/ldap/`. Las funciones de esta clase se utilizan en los archivos de muestra `UserMappingRepositoryLDAP.java` y `UserMappingSetupLDAP.java`.

Métodos públicos

abstract byte[] encrypt(byte[] plainValue)

Implemente el algoritmo de cifrado que coincida con el algoritmo de cifrado que utiliza el repositorio externo.

abstract byte[] decrypt(byte[] encryptedValue)

Implemente el algoritmo de cifrado que invierta el algoritmo de cifrado que utiliza el repositorio externo y, a continuación, devuelve la contraseña.

abstract string encode(byte[] bytes)

Escriba o implemente una función que codifique el parámetro **bytes** en una serie. Esta función codifica el valor cifrado, en bytes, en una serie.

abstract byte[] decode(String[] string)

Escriba o implemente una función que descodifique el parámetro **string** en bytes. Esta función descodifica la contraseña recuperada, una serie, en bytes para que el valor se pueda descifrar.

byte[] getBytes(char[] chars)

Esta función se hereda y puede utilizarse sin modificaciones. La función transforma cada carácter de una serie en un byte.

char[] getChars(byte[] bytes)

Esta función se hereda y puede utilizarse sin modificaciones. La función transforma cada byte en un carácter.

Atributos protegidos

SecretKey key

La clave secreta utilizada para cifrar y descifrar las contraseñas remotas.

Cipher cipher

El algoritmo que utiliza la clave secreta para cifrar la contraseña.

Clase UserMappingEntry (lenguaje de programación Java)

La clase `UserMappingEntry` es una clase de programa de utilidad que crea y conserva las opciones de correlación de usuario. Los métodos de la clase `UserMappingEntry` son llamados por las funciones `fetchUM()` y `lookupUM()` desde la clase `UserMappingRepository`.

Métodos públicos

Puede ver el uso de estas funciones en un plugin Java de muestra que recupera correlaciones de usuario de un servidor LDAP. Los archivos se encuentran en el directorio `sql1lib/samples/federated/umplugin/ldap/`. Las funciones de esta clase se utilizan en los archivos `UserMappingRepositoryLDAP.java` y `UserMappingLookupLDAP.java`.

UserMappingEntry(UserMappingRepository **repository**, String **iiInstanceName**, String **iiDatabaseName**, String **iiRemoteServerName**, String **iiAuthID**)

Este constructor se utiliza para crear una instancia en el objeto UserMappingEntry object con los parámetros de entrada siguientes:

- **iiInstance** - Nombre de instancia del servidor federado
- **iiDatabase** - Nombre de base de datos del servidor federado
- **iiRemoteServerName** - Nombre de servidor remoto para el origen de datos
- **iiAuthid** - ID de usuario local asociado con la correlación de usuario

UserMappingRepository **getRepository()**

Devuelve el objeto UserMappingRepository asociado con esta UserMappingEntry.

String **getiiInstanceName()**

Devuelve el nombre de la instancia.

String **getiiDatabaseName()**

Devuelve el nombre de la base de datos.

String **getiiRemoteServerName()**

Devuelve el nombre del servidor remoto

String **getiiAuthID()**

Devuelve el nombre del ID de usuario local asociado con la correlación de usuario.

UserMappingOption **getFirstOption()**

Devuelve el primer objeto UserMappingOption que pertenece al objeto UserMappingEntry.

void **addOption**(UserMappingOption **newOption**)

Añade un nuevo objeto UserMappingOption al objeto UserMappingEntry.

Clase UserMappingOption (lenguaje de programación Java)

La clase UserMappingOption es una clase de programa de utilidad que contiene las funciones para proporcionar al servidor federado el DI de usuario remoto y la contraseña remota.

Métodos públicos

Puede ver el uso de estas funciones en un plugin Java de muestra que recupera correlaciones de usuario de un servidor LDAP. Los archivos se encuentran en el directorio `sql1lib/samples/federated/umplugin/ldap/`. Las funciones de esta clase se utilizan en los archivos `UserMappingRepositoryLDAP.java` y `UserMappingLookupLDAP.java`.

UserMappingEntry **getEntry()**

Devuelve el objeto UserMappingEntry asociado con esta UserMappingOption.

String **getName()**

Devuelve el nombre de la opción.

void setName()

Establece el nombre de la opción.

UserMappingOption getNextOption()

Devuelve la siguiente opción.

void setNextOption(UserMappingOption nextOption)

Establece la siguiente opción.

abstract object getValue()

Devuelve el valor de la opción. Esta función debe devolver un valor de serie o un valor binario. Consulte los métodos StringOption y BinaryOption que se listan a continuación.

Clase StringOption

La clase StringOption se amplía desde la clase UserMappingOptions y contiene los métodos públicos siguientes: getValue() y setValue().

Métodos públicos**object getValue()**

Devuelve el valor de la opción de serie cuando la opción es una serie.

void setValue(String value)

Establece el valor de la opción de serie.

Clase BinaryOption

La clase BinaryOption se amplía desde la clase UserMappingOption y contiene los métodos públicos siguientes: getValue() y void setValue().

Métodos públicos**object getValue()**

Devuelve el valor de la opción binaria.

void setValue(byte[] value)

Establece el valor de la opción binaria.

Clase UserMappingException (lenguaje de programación Java)

El plugin de correlación de usuario de Java utiliza la clase UserMappingException, que es una subclase de la clase java.lang.Exception, para informar de errores.

Métodos públicos

Puede ver el uso de estas funciones en un plugin Java de muestra que recupera correlaciones de usuario de un servidor LDAP. Los archivos se encuentran en el directorio sql1lib/samples/federated/umplugin/ldap/. Las funciones de esta clase se utilizan en los archivos Java de muestra.

UserMappingException(int errorNumber)

El constructor que se utiliza para crear instancias en el objeto UserMappingException utilizado para informar de errores. El parámetro **errorNumber** que se envía al objeto UserMappingException define el tipo de error del que se informa.

int getErrorNumber()

Devuelve el número de error de la excepción.

El archivo `UserMappingRepositoryLDAP.java` contiene esta función para captar e informar de los errores cuando se prueba el plugin LDAP.

String getErrorMessage()

Devuelve el mensaje de error de la excepción.

El archivo `UserMappingRepositoryLDAP.java` contiene un método para captar e informar de los errores cuando se prueba el plugin LDAP.

Mensajes de error

La tabla siguiente lista los números de error, los nombres de constante y los mensajes de error.

Tabla 26. Números y mensajes de error

Número de error	Nombre de constante	Mensaje de error
1	INITIALIZE_ERROR	El plugin no se ha podido inicializar.
2	CONNECTION_ERROR	No se puede conectar con el repositorio.
3	AUTHENTICATION_ERROR	No se puede autenticar con el repositorio.
4	LOOKUP_ERROR	La búsqueda en el repositorio ha fallado.
5	DECRYPTION_ERROR	El descifrado ha fallado.
6	DISCONNECT_ERROR	No se pueden desconectar desde el repositorio.
7	INVALID_PARAMETER_ERROR	Parámetro no válido.
8	UNAUTHORIZED_CALLER	El interlocutor no está autorizado a llamar al plugin.

Plugin de correlación de usuario de muestra (lenguaje de programación Java)

El plugin Java de muestra recupera las entradas de usuario desde el servidor LDAP. Para utilizar el plugin en el entorno, modifique el plugin de muestra para que coincida con los valores que utiliza el servidor LDAP.

Los archivos para el plugin se encuentran en el directorio `sql1lib/samples/federated/umplugin/ldap/`. La tabla siguiente describe los archivos. Antes de modificar los archivos, cópielos en un directorio vacío. A continuación, trabaje con las copias.

Tabla 27. Descripción de los archivos del plugin de correlación de usuario de muestra (Java)

Nombre de archivo	Descripción
README.txt	Este archivo contiene una versión resumida de las instrucciones y de la documentación para probar y utilizar el plugin de muestra.

Tabla 27. Descripción de los archivos del plugin de correlación de usuario de muestra (Java) (continuación)

Nombre de archivo	Descripción
UserMappingCryptoLDAP.java	Esta clase Java contiene el código que implementa las medidas de seguridad para cifrar, descifrar, codificar y decodificar las correlaciones de usuario recuperadas desde el servidor LDAP. Debe modificar el archivo para que funcione con el servidor LDAP.
UserMappingSetupLDAP.java	Esta clase Java crea el archivo de configuración que almacena la información de conexión LDAP y otros parámetros de configuración, incluidos: dirección IP y nombre de host, SSL o no SSL, ID de usuario y contraseña.
UserMappingRepositoryLDAP.java	Esta clase Java contiene el código para conectar, desconectar y captar las correlaciones de usuario desde el servidor LDAP. El código para este archivo utiliza el esquema definido en el archivo schema.ldif. Si se cambia el esquema también se debe cambiar una sección de este archivo.
UserMappingLookupLDAP.java	Esta clase Java contiene el código para realizar una prueba de búsqueda de LDAP. Utilice este archivo para probar el plugin fuera del servidor federado. A continuación, pruebe el plugin en el servidor federado.
schema.ldif	Este archivo se carga en el servidor LDAP para definir el esquema. El archivo LDIF contiene objetos y atributos añadidos al servidor LDAP.
entry.ldif	El archivo entry.ldif añade entradas de usuario al servidor LDAP. Las entradas de usuario utilizan objetos desde el archivo schema.ldif para almacenar las correlaciones de usuario.

Desarrollo del plugin de correlación de usuario (lenguaje de programación Java)

Utilice el código de muestra como punto de inicio para desarrollar un plugin de Java que recupere las correlaciones de usuario desde un repositorio externo. El código de muestra recupera las correlaciones desde un repositorio LDAP, pero se puede modificar para acceder a cualquier repositorio externo.

Realice las verificaciones siguientes:

- El kit de desarrollo de Java (JDK) versión 1.4 o posterior está instalado.
- El archivo db2umplugin.jar. Este archivo Java está instalado como parte de la instalación del servidor DB2 o de la instalación del cliente DB2.
- Los archivos de plugin de correlación de usuario están instalados. Estos archivos, instalados como parte de la instalación del cliente DB2, están ubicados en el directorio sql1lib/samples/federated/umplugin/ldap/.
- El parámetro **java_heap_sz** se establece en 2048.
-

El plugin que desarrolle debe conectarse al repositorio externo, recuperar las correlaciones de usuario y descifrar las contraseñas remotas. El repositorio que se

utiliza determina cómo codificar el plugin. Por ejemplo, si se utiliza el repositorio LDAP que almacena contraseñas cifradas, el plugin debe contener el esquema de cifrado y la clave secreta necesarios para descodificar las contraseñas.

Tenga en cuenta que cuando envíe y utilice el plugin, enviará ID y contraseñas de usuario sensibles entre múltiples orígenes. Para proteger esta información, restrinja el acceso al código fuente del plugin y configure el programa de utilidad de auditoría de db2audit para capturar un registro VALIDATE en el archivo de registro de diagnóstico, db2diag.log, cada vez que el servidor federado utilice el plugin. El archivo de registro de diagnóstico es de gran utilidad para realizar seguimientos de uso pero también para resolver los problemas que se puedan llegar a producir.

Para desarrollar un plugin, realice las tareas siguientes:

Modificación de los archivos de plugin de correlación de usuario (lenguaje de programación Java)

El plugin Java de muestra recupera las correlaciones de usuario desde el servidor LDAP. Puede modificar el plugin para recuperar correlaciones de usuario desde otros tipos de repositorio externo. Utilice el plugin de muestra como punto de inicio para desarrollar un plugin personalizado que funcione con un repositorio externo específico.

Cada archivo de muestra desempeña una tarea distinta en el proceso de recuperación de correlaciones de usuario. Modifique varias funciones y clases en el código de muestra para trabajar con el repositorio externo específico. La tabla siguiente lista las funciones importantes.

Tabla 28. Funciones y clases que se deben modificar

Nombre de archivo	Elemento que se debe modificar	Clase a la que se debe hacer referencia
UserMappingCryptoLDAP.java	UserMappingCryptoLDAP() encrypt() decrypt() getKey() decode() encode()	Clase UserMappingCrypto Clase UserMappingException
UserMappingRepositoryLDAP.java	class UserMappingRepositoryLDAP(String configFile) connect() disconnect() fetchUM()	Clase UserMappingRepository Clase UserMappingEntry Clase UserMappingOption Clase UserMappingException
UserMappingSetupLDAP.java	Modifique este archivo para crear un archivo de configuración que almacene los valores que necesita el plugin para conectarse con el repositorio externo y recuperar las correlaciones de usuario. Si crea manualmente un archivo de configuración, asegúrese de que las contraseñas almacenadas en el archivo estén cifradas. El nombre del archivo de configuración debe coincidir con el nombre de clase del repositorio, por ejemplo, clase UserMappingRepositoryXXXX y archivo UserMappingRepositoryXXXX.cfg file.	

Modificación del archivo de muestra UserMappingCryptoLDAP (lenguaje de programación Java)

Para implementar los métodos de seguridad que utiliza el servidor LDAP, modifique las funciones que cifran, descifran, codifican y descodifican las contraseñas remotas.

Puesto que los métodos de cifrado deben ser secretos y únicos, esta tarea especifica las secciones y funciones que se deben modificar. Personalice el código para implementar los métodos de seguridad que utiliza el servidor LDAP.

Para modificar las funciones de seguridad en el archivo `UserMappingCryptoLDAP`:

1. Abra el archivo `UserMappingCryptoLDAP.java` con un editor de texto.
2. Debajo del copyright y de la declaración de limitación de responsabilidad legal de IBM, importe los paquetes a los que hace referencia el código. El plugin de muestra utiliza los paquetes `javax.crypto` y `javax.crypto.spec` de Java, que proporcionan las clases para el cifrado (cifrado y descifrado), así como los parámetros de clave y de algoritmo. Sustituya los paquetes de Java con los suyos.
3. Actualice las funciones siguientes:

public UserMappingCryptoLDAP()

Sustituya el código del cifrado con el código del cifrado que coincida con el cifrado de la contraseña que utiliza el servidor LDAP.

public byte[] encrypt(byte[] plainValue)

Esta función proporciona el código que cifra las contraseñas de manera que se puedan almacenar en el servidor LDAP. Esta función también cifra la contraseña de conexión LDAP almacenada en el archivo de configuración.

Sustituya el código de esta función con el código que cifra el parámetro `plainValue`.

public byte[] decrypt(byte[] encryptedValue)

Sustituya el código de esta función con el código que descifra el parámetro `encryptedValue`.

private SecretKey getKey()

Sustituya el código de esta función con el código que proporciona al plugin la clave utilizada para cifrar y descifrar las contraseñas.

public byte[] decode(String string)

Las contraseñas primero se cifran y, a continuación, se codifican. Esta función proporciona el código para la descodificación de contraseñas antes de que se descifren. Las contraseñas cifradas se codifican para transformar la salida binaria de la contraseña cifrada en caracteres ASCII.

Sustituya el código de esta función con el código que descodifica el parámetro `string`.

public String encode(byte[] bytes)

Las contraseñas primero se cifran y, a continuación, se codifican. Esta función proporciona el código para la codificación de la salida binaria de las contraseñas cifradas. Las contraseñas cifradas se codifican para transformar la salida binaria de la contraseña cifrada en caracteres ASCII.

Sustituya el código de esta función con el código que codifica el parámetro `bytes`.

Modificación del archivo de muestra

UserMappingRepositoryLDAP (lenguaje de programación Java)

El archivo `UserMappingRepositoryLDAP.java` del plugin de Java contiene las funciones para la conexión al servidor LDAP, la recuperación de las correlaciones

de usuario y la desconexión. Modifique el código de muestra para que coincida con las clases de objeto definidas en la estructura del directorio del servidor LDAP.

Para recuperar las correlaciones de usuario del servidor LDAP, el plugin debe buscar el directorio para las entradas de usuario con atributos que definan la correlación de usuario. La información siguiente está almacenada como atributos de usuario:

- Nombre de servidor remoto
- Nombre de instancia
- Nombre de base de datos
- Nombre de usuario remoto
- Contraseña de usuario remoto

El código de muestra asume que la entrada de usuario se identifica por la clase de objeto *inetOrgPerson* y que la entrada de correlación de usuario se identifica por la clase de objeto *IIUserMapping*. Los archivos LDIF (Lightweight Directory Interchange Format), *schema.ldif* y *entry.ldif*, cargan el esquema de muestra y las entradas de muestra en el servidor LDAP.

El código del archivo *UserMappingRepositoryLDAP.java* asume que el archivo *schema.ldif* contiene el código LDIF para los objetos y atributos con los nombres siguientes:

- Objeto *IIUserMapping*
 - Atributo *IIRemoteServerName*
 - Atributo *IIIInstanceName*
 - Atributo *IIDatabaseName*
 - Atributo *IIRemotePassword*
 - Atributo *uid*

Modifique el archivo *UserMappingRepositoryLDAP.java* para que coincida con el esquema que utiliza el servidor LDAP. El archivo *UserMappingRepositoryLDAP.java* busca y recupera las entradas de correlación de usuario del servidor LDAP. Los archivos LDIF se proporcionan como esquema de muestra y como método de muestra de almacenamiento de entradas de correlación de usuario.

Para modificar el esquema que utiliza el archivo *UserMappingRepositoryLDAP.java*:

1. Ubique el código: `private String UserObjectClassName = "inetOrgPerson"` y sustituya el valor *inetOrgPerson* con el nombre de la clase de objeto que utiliza el servidor LDAP para las entradas de usuario.
2. Opcional: Cambie los nombres de atributo que utiliza el plugin. Sustituya los valores de las variables *IIRemoteServerAttrName*, *IIIInstanceAttrName*, *IIDatabaseAttrName* y *IIRemotePasswordAttrName* con los nombres de atributo que elija.
3. Si utiliza archivos LDIF, asegúrese de que el esquema de los archivos LDIF coincida con la estructura que buscar el archivo *UserMappingRepositoryLDAP.java*.

Compilación de archivos para el plugin de correlación de usuario(lenguaje de programación Java)

Después de modificar los archivos de origen para el plugin de Java, se deben compilar.

En los siguientes mandatos, si la vía de acceso completa contiene espacios, debe delimitarla con comillas, por ejemplo "C:\program files\sql11b\java\db2umplugin.jar". %DB2PATH% es el directorio donde DB2 está instalado, por ejemplo, C:\ProgramFiles\IBM\sql11b. *directorio_instalación* es el directorio inicial de la instancia.

Los mandatos siguientes asumen que se está utilizando el convenio de denominación de archivos basado en los nombres de clases. Sustituya *xxxx* con el nombre que elija.

Para compilar los archivos de origen:

1. Emita el mandato de compilación:

Windows:

```
javac -classpath "%DB2PATH%\java\db2umplugin.jar; ^
%CLASSPATH%" -d . ^
.\UserMappingRepositoryxxxx.java ^
.\UserMappingCryptoxxxx.java ^
.\UserMappingSetupxxxx.java ^
.\UserMappingLookupXXXX.java
```

UNIX:

```
javac -classpath inst_home/sql11b/java/db2umplugin.jar:\
$CLASSPATH -d . \
./UserMappingRepositoryxxxx.java \
./UserMappingCryptoxxxx.java \
./UserMappingSetupxxxx.java \
./UserMappingLookupxxxx.java
```

2. Archive los archivos de clase Java en un archivo Java único: Un punto después del nombre del archivo de salida, dirige el mandato para que busque y ubique los archivos en el mismo directorio. Si se cambia el directorio, utilice la vía de acceso apropiada para el sistema operativo (por ejemplo, /home/user/folder o C:\test\folder).

```
jar -cfM0 UserMappingRepositoryXXXX.jar .
```

Creación del archivo de configuración para el plugin de correlación de usuario (lenguaje de programación Java)

El archivo de configuración almacena la información de conexión que utiliza el plugin deJava para conectarse al servidor LDAP.

Para crear el archivo de configuración, ejecute el programa de configuración, que le solicitará la información siguiente:

- Nombre de host o dirección IP del servidor LDAP
- Número de puerto del servidor LDAP (el valor predeterminado es 389)
- Nombre distinguido del subárbol LDAP (por ejemplo, ou=ii,o=ibm,c=us)
- ID de usuario para conectarse al servidor LDAP
- Contraseña para conectarse al servidor LDAP
- Configuración SSL

La entrada proporcionada se utiliza para crear el archivoUserMappingRepositoryLDAP.cfg, que almacena información de configuración. Si se necesita contraseña para conectarse al servidor LDAP, se cifrará con el algoritmo especificado en el archivo UserMappingCryptoLDAP.java .

En los siguientes mandatos, si la vía de acceso completa contiene espacios, debe delimitarla con comillas, por ejemplo "C:\program files\sqllib\java\db2umplugin.jar". *%DB2PATH%* es el directorio donde DB2 está instalado, por ejemplo, C:\ProgramFiles\IBM\sqllib. *directorio_instalación* es el directorio inicial de la instancia.

Para crear el archivo de configuración para el plugin LDAP de muestra:

Windows:

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^
.\UserMappingRepositoryLDAP.jar;%CLASSPATH%" UserMappingSetupLDAP
```

UNIX:

```
java -classpath inst_home/sqllib/java/db2umplugin.jar: \
./UserMappingRepositoryLDAP.jar:$CLASSPATH UserMappingSetupLDAP
```

Prueba del plugin de correlación de usuario(lenguaje de programación Java)

Para probar el plugin de Java fuera del servidor federado, desarrolle una aplicación que se conecte y recupere las correlaciones de usuario desde el repositorio externo.

Puede desarrollar un programa simple que llame al método `lookupUM()` que la clase `UserMappingRepositoryXXXX` ha heredado de la clase `UserMappingRepository` para conectarse al repositorio externo y recuperar las correlaciones de usuario. Los archivos se encuentran en el directorio `sqllib/samples/federated/umplugin/ldap/`.

En los siguientes mandatos, si la vía de acceso completa contiene espacios, debe delimitarla con comillas, por ejemplo "C:\program files\sqllib\java\db2umplugin.jar". *%DB2PATH%* es el directorio donde DB2 está instalado, por ejemplo, C:\ProgramFiles\IBM\sqllib. *directorio_instalación* es el directorio inicial de la instancia.

El programa de prueba debe utilizar los parámetros necesarios para identificar las correlaciones de usuario:

- *remoteServerName* - Nombre de servidor remoto para el origen de datos
- *iiAuthid* - ID de usuario local asociado con la correlación de usuario
- *iiInstance* - Nombre de instancia del servidor federado
- *iiDatabase* - Nombre de base de datos del servidor federado

Para probar el plugin de correlación de usuario:

Windows:

```
java -classpath "%DB2PATH%\java\db2umplugin.jar; ^
.\UserMappingRepositoryXXXX.jar;%CLASSPATH%" ^
UserMappingLookupXXXX -server nombre_servidor_remoto ^
-authid iiAuthid ^
-instance iiInstance -database iiDatabase
```

UNIX:

```
java -classpath directorio_instalación/sqllib/java/db2umplugin.jar: ^
./UserMappingRepositoryXXXX.jar:$CLASSPATH \
UserMappingLookupXXXX -server nombre_servidor_remoto \
-authid iiAuthid \
-instance iiInstance -database iiDatabase
```

Despliegue de archivos de plugin de correlación de usuario(lenguaje de programación Java)

Después de compilar y probar el plugin Java, despliegue los archivos en el servidor federado.

En los mandatos siguientes, *sqllib* es la vía de acceso completa a la instalación de DB2. Los mandatos siguientes asumen que se está utilizando el convenio de denominación de archivos basado en los nombres de clases. Sustituya *xxxx* con el nombre que elija.

Para desplegar los archivos compilados:

Copie los archivos `UserMappingRepositoryxxxx.jar` y `UserMappingRepositoryxxxx.cfg` en el directorio `sqllib/function/`.

Con los archivos en este directorio, el servidor federado podrá cargar y llamar a todas las clases de los archivos.

Configuración de acceso para el plugin de correlación de usuario(lenguaje de programación Java)

Establezca la opción `DB2_UM_PLUGIN` para configurar el servidor federado para que utilice el plugin de Java para recuperar correlaciones de usuario.

Antes de empezar

Antes de configurar el servidor federado para acceder a las correlaciones de usuario en un repositorio externo, se deben realizar las tareas siguientes:

- Desarrollar un plugin de correlación de usuario
- Desplegar el plugin en el servidor federado
- Actualizar la configuración de gestor de base de datos:

```
db2 update dbm cfg using JDK_PATH vía de acceso_jdk
db2 terminate
db2stop
db2start
```

La opción `DB2_UM_PLUGIN` debe contener la vía de acceso completa de la clase, incluido el nombre del paquete. Si desarrolla un paquete, debe incluir el nombre del paquete antes del nombre de la clase, por ejemplo, `'package.classname'`. Si se utiliza el plugin de muestra LDAP, especifique el valor `'UserMappingRepositoryLDAP'` en la opción `DB2_UM_PLUGIN`. EL plugin de muestra no está desarrollado como paquete.

Para configurar el servidor federado para que acceda al repositorio externo:

Elija cómo desea implementar el plugin de correlación de usuario:

Method	sentencia de SQL
Especifique el plugin de correlación de usuario cuando cree el derivador	<pre>CREATE WRAPPER <i>nombre de derivador</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>
Altere un derivador existente para especificar el plugin de correlación de usuario	<pre>ALTER WRAPPER <i>nombre de derivador</i> (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>

Method	sentencia de SQL
Especifique el plugin de correlación de usuario cuando cree una definición de servidor	<pre>CREATE SERVER <i>nombre_servidor</i> TYPE <i>tipo_origen_datos</i> VERSION <i>número_versión</i> WRAPPER <i>nombre de derivador</i> OPTIONS (DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>
Altere una definición de servidor existente para especificar el plugin de correlación de usuario	<pre>ALTER SERVER <i>nombre_servidor</i> OPTIONS (ADD DB2_UM_PLUGIN 'UserMappingRepositoryLDAP');</pre>

Después de establecer la opción DB2_UM_PLUGIN, el servidor federado utilizará la información de conexión especificada en el archivo UserMappingRepositoryXXXX.cfg para recuperar las correlaciones de usuario desde el repositorio externo. XXXX es el nombre del plugin.

Para alterar el derivador para que utilice un plugin distinto

```
ALTER WRAPPER nombre de derivador (
  SET DB2_UM_PLUGIN 'com.package_name.um.UserMappingRepositoryXXXX'
);
```

Para alterar la definición de servidor para que utilice un plugin distinto

```
ALTER SERVER nombre_servidor
  OPTIONS (
    SET DB2_UM_PLUGIN 'UserMappingRepositoryXXXX'
  );
```

Capítulo 34. Seguridad de Oracle en un sistema federado

Federation da soporte a un conjunto de funciones de seguridad de Oracle.

Si implementa dichas funciones en Oracle, el servidor federado podrá beneficiarse de ellas:

Seguridad de etiqueta de Oracle

El servidor federado soporta la seguridad de etiqueta de Oracle, que puede utilizar para asegurar los datos y garantizar que sólo los usuarios con la autoridad adecuada puedan verlo.

Los administradores pueden utilizar la seguridad de etiqueta de Oracle para aplicar políticas de seguridad a todas las filas de una tabla. Las políticas de seguridad determinan un nivel de usuario de acceso de datos, basándose en las autoridades otorgadas al ID de usuario o de sesión. Cuando se crea un apodo para un objeto de origen de datos de Oracle, el servidor federado detecta automáticamente si el origen de datos utiliza la seguridad de etiqueta de Oracle. Si se está utilizando la seguridad de etiqueta de Oracle, el apodo no se almacenará en la memoria caché.

Puede utilizar la sentencia ALTER NICKNAME para permitir o denegar el almacenamiento en la memoria caché. Por ejemplo, si ha creado un apodo en un objeto de origen de datos con seguridad de etiqueta de Oracle antes de que el soporte federado estuviera disponible para esta función, puede alterar el apodo para denegar el almacenamiento en la memoria caché. Si ha creado un apodo en un objeto de origen de datos con seguridad de etiqueta de Oracle y la seguridad de etiqueta de Oracle se ha eliminado, puede alterar el apodo para permitir el almacenamiento en la memoria caché.

Un administrador de base de datos puede optar por ocultar una etiqueta para impedir que algunos usuarios sepan que una fila concreta existe. En este caso, se ocultará la columna en la tabla. Los apodos que tienen columnas de etiquetas ocultas no se almacena en la memoria caché.

Autenticación proxy de Oracle y contextos fiables federados

Cree una conexión física con el origen de datos de Oracle y, a continuación, cambie la conexión a un usuario distinto en la misma conexión.

Al utilizar la autenticación proxy de Oracle y los contextos fiables federados se reduce la sobrecarga de la red producida al crear una conexión de red independiente desde el servidor federado hasta la base de datos Oracle para cada usuario, mientras se impone positivamente la identidad del usuario conectado al origen de datos Oracle. La aplicación puede cambiar de usuario a usuario, según convenga, para procesar transacciones en nombre de los usuarios.

Para configurar este caso de ejemplo, realice las tareas siguientes:

1. En el servidor Oracle, utilice la sentencia ALTER USER de Oracle para registrar a los usuarios del proxy. Se ha otorgado a Mary, la usuaria de proxy, permiso para utilizar el proxy llamado BOSS y ha obtenido el rol CLERK (asistente) para la duración de la conexión:

```
ALTER USER MARY GRANT  
CONNECT THROUGH BOSS  
WITH ROLE CLERK
```

2. En el servidor federado, cree el objeto de contexto fiable:

```
CREATE TRUSTED CONTEXT MY_FED_TCX  
BASED UPON CONNECTION USING SYSTEM AUTHID BOSS  
ATTRIBUTES (ENCRYPTION 'NONE')  
WITH USE FOR MARY WITHOUT AUTHENTICATION  
ENABLE
```

Con esta configuración, el servidor federado puede establecer una conexión fiable de extremo a extremo desde el cliente mediante el servidor federado al origen de datos de Oracle. BOSS puede establecer una conexión fiable y Mary puede reutilizarla.

Para establecer y reutilizar conexiones fiables, utilice la API que proporciona DB2.

Capítulo 35. Soporte de origen de datos para opciones federadas

Consulte esta tabla cuando desee saber si un origen de datos soporta o no una característica federada determinada.

Antes de poder utilizar algunas de estas características, es posible que necesite establecer opciones de servidor o de derivador concretas, o realizar otras tareas para habilitar la función. Para obtener más información, consulte los temas concretos de cada característica.

Tabla 29. Características y orígenes de datos soportados

Característica	Orígenes de datos
Puntos de rescate de aplicación con operaciones WRITE en apodos	DB2 para Linux, UNIX y Windows
Optimización asíncrona	Todos los orígenes de datos
Tablas de memoria caché	Familia DB2 InformixMicrosoft SQL ServerOracleSybase
Importación de datos a apodos	Familia DB2 InformixMicrosoft SQL ServerOracleSybaseTeradata
Tolerancia a errores en expresiones de tablas anidadas	Familia DB2 InformixJDBCMicrosoft SQL ServerODBCOracleSybaseTeradata
Repositorio de correlación de usuarios externos	Todos los orígenes de datos
Indicadores de salud federados	Familia DB2 Excel InformixJDBCMicrosoft SQL ServerODBCOracleSybase Archivos con estructura de tablaTeradataXML (sólo apodos raíz)
Procedimientos federados	Familia DB2, en modalidad fiable Oracle, en modalidad fiable Microsoft SQL Server, en modalidad fiable Sybase, en modalidad delimitada, con el servidor federado instalado en UNIX Sybase, en modalidad delimitada o fiable, con el servidor federado instalado en Linux o Microsoft Windows
Contextos fiables federados	DB2 para Linux, UNIX y Windows Versión 9.5 DB2 para z/OS Versión 9 Oracle
Proxy HTTP	Servicios Web XML
Aislamiento a nivel de conexión	Familia DB2 InformixJDBCMicrosoft SQL ServerODBCOracleSybase
Control de acceso basado en etiquetas	DB2 para Linux, UNIX y Windows Versión 9.1 y 9.5 Oracle
Operaciones LOB de lectura y escritura	DB2 para z/OS DB2 para Linux, UNIX y Windows DB2 para System i Oracle

Tabla 29. Características y orígenes de datos soportados (continuación)

Característica	Orígenes de datos
Operaciones LOB de sólo lectura	BioRSInformixJDBCMicrosoft SQL ServerODBCScriptSybaseTeradataServicios Web XML
Tablas de consulta materializadas	Todos los orígenes de datos, con restricciones específicas
Recurso de actualización de estadísticas de apodo	BioRS Familia DB2 Excel InformixJDBCMicrosoft SQL ServerODBCOracleSybase Archivos con estructura de tablaTeradataXML (sólo apodos raíz)
Sesiones de paso a través	DRDA InformixOracleMicrosoft SQL ServerSybaseTeradata
Tipo de datos XML remotos	DB2 para Linux, UNIX y Windows Derivador XML
Proxy SOCKS	BioRSScriptServicios Web XML
Capa de sockets seguros (SSL)	Servicios Web XML
Aislamiento a nivel de sentencia	Familia DB2 Microsoft SQL Server
Transacciones de asignación de dos fases	DB2 para Linux, UNIX y Windows, en modalidad fiable DB2 para System i, en modalidad fiable DB2 para z/OS, en modalidad fiable Informix, en modalidad fiable Microsoft SQL Server, en modalidad fiable, con el servidor federado instalado en Microsoft Windows Oracle, en modalidad fiable Sybase, en modalidad fiable, con el servidor federado instalado en Microsoft Windows Sybase, en modalidad delimitada, con el servidor federado instalado en UNIX
Soporte para Unicode	Todos los orígenes de datos

Capítulo 36. Guía de consulta de opciones para orígenes de datos

Cada origen de datos soporta determinadas opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Información de consulta sobre opciones de BioRS

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse en las sentencias CREATE WRAPPER y CREATE SERVER.

Tabla 30. Opciones de derivador para BioRS

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Esta opción es obligatoria si el valor de PROXY_TYPE es HTTP o SOCKS. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.

Tabla 30. Opciones de derivador para BioRS (continuación)

Nombre	Descripción
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Esta opción es obligatoria si el valor de PROXY_TYPE es HTTP o SOCKS. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE. Si establece esta opción en HTTP o SOCKS, también deberá especificar las opciones PROXY_SERVER_NAME y PROXY_SERVER_PORT.

Opciones de servidor

Tabla 31. Opciones de servidor para BioRS

Nombre	Descripción
CASE_SENSITIVE	Especifica si el servidor BioRS trata los nombres distinguiendo entre mayúsculas y minúsculas. Los valores válidos son Y y N. El valor predeterminado es Y (los nombres se tratan distinguiendo entre mayúsculas y minúsculas). En el producto BioRS, un parámetro de configuración controla si hay que distinguir entre mayúsculas y minúsculas en los datos almacenados en el servidor BioRS. La opción CASE_SENSITIVE y el parámetro de configuración deben especificar el mismo valor. Si tiene que cambiar el valor de la opción CASE_SENSITIVE después de crear la definición de servidor, deberá descartar la definición de servidor y crearla de nuevo, así como todos los apodos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 31. Opciones de servidor para BioRS (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
NODE	Obligatoria. Especifica el nombre de host DNS o la dirección IP del sistema en que está disponible la herramienta de consultas de BioRS. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6. El valor predeterminado es localhost.
PORT	Especifica el puerto de conexión del servidor BioRS. Los valores válidos son un puerto numérico o un nombre de servicio TCP/IP. El valor predeterminado es 5014.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
TIMEOUT	Especifica el tiempo máximo, en minutos, que el servidor federado esperará una respuesta del servidor remoto. El valor predeterminado es 10.

Opciones de correlación de usuarios

Tabla 32. Opciones de correlación de usuarios para BioRS

Opción	Descripción
GUEST	Especifica que se utiliza el ID de autenticación GUEST de BioRS para realizar operaciones. Los valores válidos son Y y N. El valor predeterminado es N (el ID GUEST de BioRS no se utiliza). Esta opción no es válida si se especifican las opciones REMOTE_AUTHID y REMOTE_PASSWORD.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de apodo

Tabla 33. Opciones de apodo para BioRS

Opción	Descripción
REMOTE_OBJECT	Especifica el nombre del banco de datos de BioRS asociado con el apodo. Este nombre determina el esquema y el banco de datos de BioRS del apodo. El nombre también especifica la relación del apodo con otros apodos. El que esta opción distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. No se puede utilizar la sentencia ALTER NICKNAME para cambiar o suprimir este nombre. Si el nombre del banco de datos de BioRS cambia, deberá suprimir el apodo y crearlo de nuevo.
TIMEOUT	Especifica el tiempo máximo, en minutos, que se esperará una respuesta del servidor de orígenes de datos. El valor predeterminado es 10.

Opciones de columna

Tabla 34. Opciones de columna para BioRS

Opción	Descripción
ELEMENT_NAME	Especifica el nombre de elemento de BioRS. El que este nombre distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. Debe especificar el nombre de elemento de BioRS solamente si es distinto del nombre de columna.
IS_INDEXED	Especifica si la columna correspondiente está indexada y, por lo tanto, si puede hacerse referencia a ella en un predicado. Los valores válidos son Y y N. El valor predeterminado es N (la columna no está indexada).
REFERENCED_OBJECT	Especifica el nombre del banco de datos de BioRS al que hace referencia la columna actual. El que este nombre distinga entre mayúsculas y minúsculas depende de que lo haga el servidor BioRS y del valor de la opción de servidor CASE_SENSITIVE. Esta opción sólo es válida para las columnas que tienen el tipo de datos Reference de BioRS.

Información de consulta sobre opciones de bases de datos de DB2

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a orígenes de datos DB2 y se identifican las opciones obligatorias que deben especificarse.

Tabla 35. Opciones de derivador para orígenes de datos DB2

Name	Description
DB2_FENCED	Necesario. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 35. Opciones de derivador para orígenes de datos DB2 (continuación)

Name	Description
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 36. Opciones de servidor para orígenes de datos DB2

Name	Description
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 36. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
DATE_COMPAT	Especifica si se aplica el parámetro <code>date_compat</code> a la base de datos. Los valores válidos son Y y N. El valor predeterminado es N. Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.7 or posterior.
DBNAME	Necesario. Especifica la base de datos concreta a utilizar para la conexión de base de datos DB2 remota inicial. Esta base de datos concreta es el alias de base de datos para la base de datos DB2 remota catalogada en el servidor federado mediante el mandato <code>CATALOG DATABASE</code> o el asistente de configuración de DB2.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, <code>"UserMappingRepositoryLDAP"</code> . En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.

Tabla 36. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.

Tabla 36. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
NO_EMPTY_STRING	<p>Especifica si el servidor de orígenes de datos remotos contiene series vacías. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para orígenes de datos Oracle remotos, el valor predeterminado es Y; todos los valores de serie vacíos se convierten a valores NULL. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos puede contener series vacías.</p> <p>Puede mejorar el rendimiento de su sistema estableciendo esta opción a Y en las configuraciones del sistema en las que el servidor federado está en modalidad compatible con VARCHAR2, pero el origen de datos remoto no es compatible con VARCHAR2.</p>
NUMBER_COMPAT	<p>Especifica si el servidor de orígenes de datos admite el tipo de datos NUMBER. Los valores válidos son Y y N. El valor predeterminado es N; el servidor de orígenes de datos no admite el tipo de datos NUMBER. En los sistemas en los que el servidor federado no admite el tipo de datos NUMBER pero el servidor de orígenes de datos sí, debe establecer la opción NUMBER_COMPAT a Y porque el servidor de orígenes de datos puede devolver resultados DECFLOAT que están fuera del intervalo del tipo de datos DECIMAL y causan el error SQLSTATE 560BD.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.7 y posterior.</p>
OLD_NAME_GEN	<p>Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.</p>

Tabla 36. Opciones de servidor para orígenes de datos DB2 (continuación)

Name	Description
PUSHDOWN	<p>Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.</p>
SAME_DECFLT_ROUNDING	<p>Especifica si la modalidad de redondeo del servidor federado y del servidor de orígenes de datos está utilizando la misma configuración de modalidad de redondeo DECFLOAT. Los valores válidos son Y y N. El valor predeterminado es N; el servidor federado y el servidor remoto tienen configuraciones de modalidad de redondeo DECFLOAT diferentes.</p> <p>Importante: Si establece esta opción a Y en los casos en que las modalidades de redondeo son distintas en el servidor federado y el servidor de orígenes de datos, es posible que reciba resultados de redondeo DECFLOAT incorrectos.</p> <p>Para configurar un servidor federado y un servidor de orígenes de datos existentes que utilizan la misma configuración de modalidad de redondeo DECFLOAT, utilice la sentencia ALTER SERVER.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior.</p>
VARCHAR2_COMPAT	<p>Especifica si el origen de datos remoto es compatible con VARCHAR2. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para los orígenes de datos Oracle remotos, el valor predeterminado es Y; el origen de datos es compatible con VARCHAR2. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos no está establecido en modalidad compatible VARCHAR2.</p> <p>Debe establecer esta opción de servidor a Y si su origen de datos DB2 Database para Linux, UNIX y Windows, ODBC o JDBC está configurado en modalidad compatible con VARCHAR2.</p>

Opciones de correlación de usuarios

Tabla 37. Opciones de de correlación de usuarios para orígenes de datos DB2

Opción	Description
FED_PROXY_USER	<p>Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Si especifica la opción de correlación de usuarios FED_PROXY_USER, también deberá especificar la opción de servidor FED_PROXY_USER.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.</p>
ACCOUNTING_STRING	<p>Necesario en caso que haya que pasar información contable. Especifica una serie contable DRDA. Los valores válidos incluyen cualquier serie que tenga 255 o menos caracteres.</p>
REMOTE_AUTHID	<p>Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.</p>
REMOTE_PASSWORD	<p>Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.</p>
USE_TRUSTED_CONTEXT	<p>Especifica si la correlación de usuarios es fiable. Los valores válidos son Y y N. El valor predeterminado es N (la correlación de usuarios no es fiable y sólo puede utilizarse en conexiones federadas de salida que no son fiables). Y especifica que la correlación de usuarios es fiable y que puede utilizarse en conexiones federadas de salida fiables y no fiables.</p> <p>Restricción: Esta opción de servidor sólo es válida para DB2 Database para Linux, UNIX y Windows, Versión 9.5 y posterior, y DB2 for z/OS Versión 9 y posterior.</p>

Opciones de columna

Tabla 38. Opciones de columna para orígenes de datos DB2

Opción	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
NO_EMPTY_STRING	Especifica si el servidor de orígenes de datos remotos contiene series vacías. Los valores válidos son Y y N. El valor predeterminado varía en función del origen de datos remoto. Para orígenes de datos Oracle remotos, el valor predeterminado es Y; todos los valores de serie vacíos se convierten a valores NULL. Para el resto de orígenes de datos remotos, el valor predeterminado es N; el origen de datos puede contener series vacías.
XML_ROOT	Especifica el elemento raíz XML que se debe añadir a los valores de una columna XML que hace referencia a una secuencia XML. Esta opción asegura que los valores de la columna XML son un documento XML con el formato correcto.

Información de consulta sobre opciones de Excel

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor y de apodo.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 39. Opciones de derivador para Excel

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Opciones de servidor

Tabla 40. Opciones de servidor para Excel

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Opciones de apodo

Tabla 41. Opciones de apodo para Excel

Opción	Descripción
FILE_PATH	Obligatoria. Especifica la vía de acceso totalmente calificada del directorio y el nombre de archivo de la hoja de cálculo Excel a la que desea acceder.
RANGE	Especifica el rango de celdas que se va a utilizar; por ejemplo, A1:C100. El valor que precede a los dos puntos especifica la celda superior izquierda del rango. El valor que sigue a los dos puntos especifica la celda inferior derecha del rango.

Información de consulta sobre opciones de Informix

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 42. Opciones de derivador para Informix

Nombre	Descripción
DB2_FENCED	Necesario. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 43. Opciones de servidor para Informix

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N y I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.

Tabla 43. Opciones de servidor para Informix (continuación)

Nombre	Descripción
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DBNAME	Necesario. Especifica el nombre de la base de datos Informix a la que desea acceder.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos. Si más de un plan de acceso cumple estos criterios, se elegirá el que tenga el coste menor.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.

Tabla 43. Opciones de servidor para Informix (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
INFORMIX_CLIENT_LOCALE	Especifica qué CLIENT_LOCALE utilizar para la conexión entre el servidor federado y el servidor de orígenes de datos. El valor es cualquier entorno local Informix válido. Si no especifica esta opción, la variable de entorno CLIENT_LOCALE se establece en el valor especificado en el archivo db2dj.ini. Si el archivo db2dj.ini no especifica la variable de entorno CLIENT_LOCALE, el INFORMIX_CLIENT_LOCALE se establece en el entorno local de Informix más parecido a la página de códigos y al territorio de la base de datos federada.

Tabla 43. Opciones de servidor para Informix (continuación)

Nombre	Descripción
INFORMIX_DB_LOCALE	Especifica qué DB_LOCALE de Informix utilizar para la conexión entre el servidor federado y el servidor de orígenes de datos. Si no se especifica la opción INFORMIX_DB_LOCALE, la variable de entorno local DB_LOCALE de Informix se establece en el valor especificado en el archivo db2dj.ini. Si el archivo db2dj.ini no especifica un valor, no se establece la variable de entorno DB_LOCALE de Informix.
INFORMIX_LOCK_MODE	Especifica qué modalidad de bloqueo establecer para un origen de datos de Informix. El derivador de Informix emite el mandato SET LOCK MODE inmediatamente después de conectarse a un origen de datos de Informix. Los valores válidos son W, N y un número. El valor predeterminado es W; el derivador espera de forma indefinida a que se libere el bloqueo. N especifica no esperar; se devuelve un error inmediatamente. Utilice un número para especificar el tiempo máximo, en segundos, que debe esperar. Si se produce un tiempo muerto o se excede el tiempo de espera, utilice la sentencia ALTER SERVER para cambiar el valor de la opción INFORMIX_LOCK_MODE. Por ejemplo: <pre>ALTER SERVER TYPE informix VERSION 9 WRAPPER informix OPTIONS (ADD informix_lock_mode '60')</pre>
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.

Tabla 43. Opciones de servidor para Informix (continuación)

Nombre	Descripción
IUD_APP_SVPT_ENFORCE	Especifica si el servidor federado debe aplicar el uso de sentencias de punto de recuperación de aplicación. Los valores válidos son Y y N. El valor predeterminado es Y; en caso que el origen de datos no aplique sentencias de punto de recuperación de aplicación y que ocurra un error durante la operación de inserción, actualización o supresión, el servidor federado deshace la transacción y se devuelve el código de error SQL1476N. Es recomendable que utilice el valor predeterminado.
NODE	Necesario. Especifica el nombre por el que el origen de datos está definido como una instancia a su sistema de gestión de bases de datos relacionales.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Opciones de correlación de usuarios

Tabla 44. Opciones de correlación de usuarios para Informix

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.

Tabla 44. Opciones de correlación de usuarios para Informix (continuación)

Nombre	Descripción
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 45. Opciones de columna para Informix

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de JDBC

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 46. Opciones de derivador para JDBC

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. El único valor válido es Y porque el servidor DB2 solamente permite cargar la JVM en modalidad delimitada. El valor predeterminado es Y (el derivador se ejecuta en modalidad delimitada).

Tabla 46. Opciones de derivador para JDBC (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 47. Opciones de servidor para JDBC

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 47. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
DATEFORMAT	Especifica el formato de fecha que utiliza el origen de datos. Utilice 'DD', 'MM' y 'YY' o 'YYYY' para especificar el formato de fecha. Puede especificar un delimitador, como por ejemplo un espacio, un guión o una coma. Por ejemplo, el formato 'YYYY-MM-DD' especifica una fecha del tipo 1958-10-01. El valor puede contener valores nulos.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 0. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 47. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
DRIVER_CLASS	<p>Especifica la biblioteca de controladores JDBC. El servidor se puede registrar en varios orígenes de datos JDBC si el controlador JDBC se ajusta a la especificación JDBC versión 3.0 y posteriores. Consulte la documentación del controlador JDBC para obtener más información sobre las especificaciones JDBC y sobre cómo establecer la opción de servidor DRIVER_CLASS.</p> <p>Ejemplo</p> <p>En este ejemplo, se especifica la biblioteca de controladores JDBC com.ibm.db2.jcc.DB2Driver:</p> <pre>DRIVER_CLASS 'com.ibm.db2.jcc.DB2Driver'</pre> <p>Importante: Si especifica esta opción, también deberá especificar la opción de servidor URL.</p>
DRIVER_PACKAGE	<p>Especifica los paquetes de controladores JDBC. Utilice un separador de vías de acceso para especificar varios paquetes de clases de controlador. Utilice un punto y coma en los sistemas operativos Windows y dos puntos en los sistemas operativos Linux y Unix.</p> <p>Ejemplo</p> <p>En este ejemplo, se especifican varios paquetes de controladores separados por dos puntos en sistemas operativos Linux:</p> <pre>DRIVER_PACKAGE '/via1/archivo1.jar: /via2/archivo2.jar'</pre>
FOLD_ID	<p>Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>
FOLD_PW	<p>Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>

Tabla 47. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
IO_RATIO	<p>Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23}. El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.</p>
JDBC_LOG	<p>Especifica si el derivador JDBC crea archivos de registro para el rastreo de errores. Los valores válidos son Y y N. El valor predeterminado es N (el archivo de registro no se crea). Si esta opción de servidor se establece en Y, el derivador JDBC graba archivos de registro JDBC en el archivo <code>jdbc_wrapper_id_prod.log</code>, donde <code>id_prod</code> es el ID de producto. El archivo de registro se almacena en el directorio especificado por el parámetro de configuración del gestor de bases de datos DB2 DIAGPATH. El directorio predeterminado en sistemas UNIX es <code>inst_home/sql/lib/db2dump</code>.</p> <p>Recomendación: Establecer esta opción de servidor en YES (SÍ) afectará al rendimiento del sistema y no debe habilitar el registro de anotaciones en sistemas de producción.</p>
OLD_NAME_GEN	<p>Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.</p>

Tabla 47. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.
TIMEFORMAT	Especifica el formato de hora que utiliza el origen de datos. Utilice 'hh12', 'hh24', 'mm', 'ss', 'AM' y 'A.M.' para especificar el formato de hora. Por ejemplo, el formato 'hh24:mm:22' especifica una hora del tipo 16:00:00. El formato 'hh12:mm:ss AM' especifica una hora del tipo 8:00:00 AM. El valor puede contener valores nulos.
TIMESTAMPFORMAT	Especifica el formato de indicación de fecha y hora que utiliza el origen de datos. Los valores válidos son los que utilizan las opciones DATEFORMAT y TIMEFORMAT. Especifique 'n' para las décimas de segundo, 'nn' para las centésimas de segundo, 'nnn' para los milisegundos y así hasta 'nnnnnn' para los microsegundos. Por ejemplo, el formato 'YYY-MM-DD-hh24:mm:ss.nnnnnn' especifica una indicación de fecha y hora del tipo 1994-01-01-24:00:00.000000. El valor puede contener valores nulos.

Tabla 47. Opciones de servidor para JDBC (continuación)

Nombre	Descripción
URL	<p>Especifica la serie de conexión JDBC del servidor remoto.</p> <p>La serie de conexión JDBC consta de tres partes separadas por dos puntos:</p> <ul style="list-style-type: none"> • Protocolo de la base de datos • Nombre del tipo de la base de datos o nombre del controlador de conectividad • Identidad de la base de datos mediante un alias o un subnombre <p>Ejemplo</p> <p>En este ejemplo, la serie de conexión JDBC es</p> <pre>jdbc:db2://cn.ibm.com:50471/testdb:</pre> <p>URL</p> <pre>'jdbc:db2://cn.ibm.com:50471/testdb'</pre> <p>El servidor se puede registrar en varios orígenes de datos JDBC si el controlador JDBC se ajusta a la especificación JDBC versión 3.0 y posteriores. Consulte la documentación del controlador JDBC para obtener más información sobre las especificaciones JDBC y sobre cómo establecer la opción de servidor URL.</p> <p>Importante: Si especifica esta opción, también deberá especificar la opción de servidor DRIVER_CLASS.</p>
VARCHAR_NO_TRAILING_BLANKS	<p>Especifica si el origen de datos contiene columnas VARCHAR que contienen como mínimo un carácter en blanco de cola. El valor predeterminado es N (las columnas VARCHAR contienen como mínimo un carácter en blanco de cola).</p>

Opciones de correlación de usuarios

Tabla 48. Opciones de correlación de usuarios para JDBC

Nombre	Descripción
REMOTE_AUTHID	<p>Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.</p>
REMOTE_PASSWORD	<p>Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.</p>

Opciones de columna

Tabla 49. Opciones de columna para JDBC

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si hay como mínimo un blanco de cola en la columna VARCHAR.

Información de consulta sobre opciones de Microsoft SQL Server

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 50. Opciones de derivador para Microsoft SQL Server

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 50. Opciones de derivador para Microsoft SQL Server (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 51. Opciones de servidor para Microsoft SQL Server

Nombre	Descripción
CODEPAGE	Especifica la página de códigos que corresponde al juego de caracteres codificado de la configuración de cliente del origen de datos. En sistemas UNIX y Microsoft Windows que utilizan una base de datos federada que no es Unicode, el valor predeterminado es la página de códigos de la base de datos federada. En sistemas UNIX que utilizan una base de datos federada Unicode, el valor predeterminado es 1208. En sistemas Windows que utilizan una base de datos federada Unicode, el valor predeterminado es 1202.
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.

Tabla 51. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
COMM_RATE	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la velocidad de CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4.
DBNAME	Obligatoria. Especifica el alias de la base de datos a la que desea acceder. El valor es sensible a mayúsculas y minúsculas.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos. Si más de un plan de acceso cumple estos criterios, se elegirá el que tenga el coste menor.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 51. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
DB2_TWO_PHASE_COMMIT	<p>Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.</p> <p>Importante: Si establece esta opción en Y, también deberá especificar la opción XA_OPEN_STRING_OPTION.</p>
DB2_UM_PLUGIN	<p>Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.</p>
DB2_UM_PLUGIN_LANG	<p>Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.</p>
FOLD_ID	<p>Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>
FOLD_PW	<p>Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.</p>

Tabla 51. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Si el servidor federado utiliza Microsoft Windows, el valor de NODE es el nombre DSN del sistema especificado para Microsoft SQL Server. Si el servidor federado utiliza UNIX o Linux, el valor de NODE se define en el archivo <code>odbc.ini</code> . El valor es sensible a mayúsculas y minúsculas.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PASSWORD	Especifica si hay que enviar contraseñas a un origen de datos o no. El valor predeterminado es Y.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Tabla 51. Opciones de servidor para Microsoft SQL Server (continuación)

Nombre	Descripción
XA_OPEN_STRING_OPTIONS	Opción obligatoria si se establece DB2_TWO_PHASE_COMMIT en Y. Especifica el ID del gestor de recursos del Registro de Microsoft SQL Server.

Opciones de correlación de usuarios

Tabla 52. Opciones de correlación de usuarios para Microsoft SQL Server

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 53. Opciones de columna para Microsoft SQL Server

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de ODBC

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 54. Opciones de derivador para ODBC

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable). Importante: Si establece esta opción en Y para un sistema UNIX, también deberá establecer la opción de derivador DB2_SOURCE_CLIENT_MODE.
DB2_SOURCE_CLIENT_MODE	Especifica que el cliente del origen de datos es de 32 bits y que la instancia de base de datos del servidor federado es de 64 bits. El único valor válido es 32 bits. Esta opción sólo es válida para UNIX. Importante: Si establece esta opción, también deberá establecer la opción de derivador DB2_FENCED en Y.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
MODULE	Obligatoria para servidores federados que se ejecutan en un sistema UNIX. Especifica la vía de acceso completa de la biblioteca que contiene la implementación del gestor de controladores ODBC o la implementación de SQL/CLI. No hay valor predeterminado para UNIX. En un sistema Microsoft Windows, el valor predeterminado es odbc32.dll.

Opciones de servidor

Tabla 55. Opciones de servidor para ODBC

Nombre	Descripción
CODEPAGE	Especifica la página de códigos que corresponde al juego de caracteres codificado de la configuración de cliente del origen de datos. En sistemas UNIX y Windows que utilizan una base de datos federada que no es Unicode, el valor predeterminado es la página de códigos que utiliza la base de datos federada. En sistemas UNIX que utilizan una base de datos federada Unicode, el valor predeterminado es 1208. En sistemas Windows que utilizan una base de datos federada Unicode, el valor predeterminado es 1202.
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DATEFORMAT	Especifica el formato de fecha que utiliza el origen de datos. Utilice 'DD', 'MM' y 'YY' o 'YYYY' para especificar el formato de fecha. Puede especificar un delimitador, como por ejemplo un espacio, un guión o una coma. Por ejemplo, el formato 'YYYY-MM-DD' especifica una fecha como 1958-10-01. El valor puede contener valores nulos.

Tabla 55. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
DBNAME	Especifica el nombre de la base de datos de origen de datos a la que desea acceder.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 0. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.

Tabla 55. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre del nodo o el nombre DSN del sistema que se asigna al origen de datos ODBC cuando se define el DSN. El valor es sensible a mayúsculas y minúsculas.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombres de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Tabla 55. Opciones de servidor para ODBC (continuación)

Nombre	Descripción
TIMEFORMAT	Especifica el formato de hora que utiliza el origen de datos. Utilice 'hh12', 'hh24', 'mm', 'ss', 'AM' y 'A.M.' para especificar el formato de hora. Por ejemplo, el formato 'hh24:mm:22' especifica una hora del tipo 16:00:00. El formato 'hh12:mm:ss AM' especifica una hora del tipo 8:00:00 AM. El valor puede contener valores nulos.
TIMESTAMPFORMAT	Especifica el formato de indicación de fecha y hora que utiliza el origen de datos. Los valores válidos son los que utilizan las opciones DATEFORMAT y TIMEFORMAT. Especifique 'n' para las décimas de segundo, 'nn' para las centésimas de segundo, 'nnn' para los milisegundos y así hasta 'nnnnnn' para los microsegundos. Por ejemplo, el formato 'YYY-MM-DD-hh24:mm:ss.nnnnnn' especifica una indicación de fecha y hora del tipo 1994-01-01-24:00:00.000000. El valor puede contener valores nulos.
VARCHAR_NO_TRAILING_BLANKS	Especifica si el origen de datos contiene columnas VARCHAR que contienen como mínimo un carácter en blanco de cola. El valor predeterminado es N (las columnas VARCHAR contienen como mínimo un carácter en blanco de cola).

Opciones de correlación de usuarios

Tabla 56. Opciones de correlación de usuarios para ODBC

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 57. Opciones de columna para ODBC

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si hay como mínimo un blanco de cola en la columna VARCHAR.

Información de consulta sobre opciones de Oracle

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 58. Opciones de derivador para Oracle

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Tabla 58. Opciones de derivador para Oracle (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 59. Opciones de servidor para Oracle

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.

Tabla 59. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones de salida fiables cuando la conexión de entrada no es fiable. Importante: El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD.

Tabla 59. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre del nodo en que reside el servidor de bases de datos Oracle. Obtenga el nombre del nodo del archivo tnsnames.ora.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.

Tabla 59. Opciones de servidor para Oracle (continuación)

Nombre	Descripción
PLAN_HINTS	Especifica si se van a habilitar las sugerencias de plan o no. Las sugerencias de plan son fragmentos de sentencias que proporcionan información adicional que el optimizador de orígenes de datos utiliza para mejorar el rendimiento de la consulta. El optimizador de orígenes de datos utiliza las sugerencias de plan para decidir si utilizar un índice o no y qué índice o qué secuencia de unión de tabla utilizar. Los valores válidos son Y y N. El valor predeterminado es N (las sugerencias de plan no se habilitan). Y especifica que las sugerencias de plan se van a habilitar en el origen de datos si éste admite las sugerencias de plan.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el servidor federado permite que el origen de datos evalúe operaciones). N especifica que el servidor federado recupera columnas del origen de datos remoto.
VARCHAR_NO_TRAILING_BLANKS	Especifica para un servidor determinado si las columnas VARCHAR contienen blancos de cola. Para aplicar la opción a una columna individual, utiliza la opción de columna VARCHAR_NO_TRAILING_BLANKS.
XA_OPEN_STRING_OPTIONS	Especifica información adicional que hay que añadir a la serie. Por ejemplo, la información podría ser el directorio de los archivos de rastreo.

Opciones de correlación de usuarios

Tabla 60. Opciones de correlación de usuarios para Oracle

Nombre	Descripción
FED_PROXY_USER	Especifica el ID de autorización que hay que utilizar para establecer todas las conexiones fiables de salida cuando la conexión de entrada no es fiable. Importante: El usuario cuyo ID se especifica en esta opción debe tener una correlación de usuarios que especifique las opciones REMOTE_AUTHID y REMOTE_PASSWORD. Si especifica la opción de correlación de usuarios FED_PROXY_USER, también deberá especificar la opción de servidor FED_PROXY_USER.

Tabla 60. Opciones de correlación de usuarios para Oracle (continuación)

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.
USE_TRUSTED_CONTEXT	Especifica si la correlación de usuarios es fiable. Los valores válidos son Y y N. El valor predeterminado es N (la correlación de usuarios no es fiable y sólo puede utilizarse en conexiones de salida que no son fiables). Y especifica que la correlación de usuarios es fiable y que puede utilizarse en conexiones de salida fiables y no fiables.

Opciones de columna

Tabla 61. Opciones de columna para Oracle

Nombre	Descripción
NUMERIC_STRING	Especifica si la columna contiene series de caracteres numéricos que incluyen blancos. Los valores válidos son Y y N. El valor predeterminado es N (la columna no contiene series numéricas que incluyen blancos). Si la columna contiene solamente series numéricas seguidas por blancos de cola, no especifique Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.
VARCHAR_NO_TRAILING_BLANKS	Especifica si la columna contiene blancos de cola.

Información de consulta sobre opciones de Script

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 62. Opciones de derivador para scripts

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.

Opciones de servidor

Tabla 63. Opciones de servidor para scripts

Nombre	Descripción
DAEMON_PORT	Especifica el número de puerto en que el daemon Script está a la escucha de solicitudes de trabajos de scripts. El valor predeterminado es 4099. El número de puerto debe ser el mismo que el de la opción DAEMON_PORT en el archivo de configuración del daemon. Si se configura un nombre de servicio para el daemon Script, utilice un nombre de servicio TCP/IP.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 63. Opciones de servidor para scripts (continuación)

Nombre	Descripción
NODE	Obligatoria. Especifica el nombre de host DNS o la dirección IP del sistema en que se ejecuta el daemon Script. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE y SOCKS. El valor predeterminado es NONE.

Opciones de correlación de usuarios

Tabla 64. Opciones de correlación de usuarios para scripts

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.

Opciones de apodo

Tabla 65. Opciones de apodo para scripts

Nombre	Descripción
DATASOURCE	Opción obligatoria para el apodo raíz. Especifica el nombre del script que hay que llamar. El script que especifique como el valor de esta opción también debe especificarse en el archivo de configuración del daemon Script. Importante: Esta opción sólo es válida para el apodo raíz.
NAMESPACES	Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis: <pre>NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"'</pre> Utilice comas para separar varios espacios de nombres. Por ejemplo: <pre>NAMESPACES='http://www.miweb.com/cliie", i='http://www.miweb.com/cliie/id", n='http://www.miweb.com/cliie/nombre"'</pre>
STREAMING	Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.
TIMEOUT	Especifica el tiempo máximo, en minutos, que se esperará algún resultado del daemon. El valor predeterminado es 60. Esta opción sólo es válida para el apodo raíz.
VALIDATE	Especifica si el documento fuente se valida para asegurarse de que se ajusta a un esquema XML o a una definición de tipo de documento (DTD) antes de que los datos se extraigan de él. El valor predeterminado es N (la validación no se realiza). Antes de establecer el valor en Y, el archivo de esquema o el archivo DTD ha de estar en la ubicación especificada por el documento fuente. Esta opción sólo es válida para el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.

Tabla 65. Opciones de apodo para scripts (continuación)

Nombre	Descripción
XPATH	Especifica la expresión XPath que identifica los elementos XML que representan tuplas individuales. La opción de apodo XPATH para un apodo hijo se evalúa en el contexto de la vía de acceso especificada por la opción de apodo XPATH de su padre. La expresión XPath se utiliza como contexto para evaluar valores de columnas identificadas por la presencia de la opción de columna XPATH.

Opciones de columna

Tabla 66. Opciones de columna para scripts

Nombre	Descripción
DEFAULT	Especifica un valor predeterminado para una columna de entrada de script. Si la consulta SQL no proporciona un valor, se utilizará este valor predeterminado.
FOREIGN_KEY	Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.
INPUT_MODE	Especifica la modalidad de entrada de la columna. Los valores válidos son CONFIG y FILE_INPUT. CONFIG trata el valor como la modalidad de entrada de una columna. FILE_INPUT especifica un archivo que almacena el valor. El derivador pasa el valor especificado al daemon Script.

Tabla 66. Opciones de columna para scripts (continuación)

Nombre	Descripción
POSITION	Especifica un valor entero para parámetros posicionales. Si el valor posicional se establece en un entero, esta entrada deberá estar en esta posición en la línea de mandatos. Si se establece esta opción, el conmutador se inserta en la ubicación adecuado cuando se ejecuta la consulta. Si se establece POSITION en -1, la opción se añade como la última opción de la línea de mandatos. Un valor entero POSITION no puede utilizarse dos veces en el mismo apodo. Esta opción sólo se aplica a columnas de entrada.
PRIMARY_KEY	Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.
SWITCH	Especifica un distintivo para el script en la línea de mandatos. El valor de esta opción precedía el valor de columna suministrado por WSSCRIPT.ARGS o por el valor predeterminado, si existe. Si no especifica un valor para esta opción y existe un valor predeterminado para la columna, éste se añade sin información de conmutador. Esta opción es obligatoria para las columnas de entrada.
SWITCH_ONLY	Permite utilizar conmutadores sin un argumento de línea de mandatos. Los valores válidos son Y y N. Si establece esta opción en Y y los valores de entrada válidos serán Y y N. Si el valor de entrada es Y, sólo se añade el conmutador a la línea de mandatos. Si el valor de entrada es N, no se añade ningún valor a la línea de mandatos.
VALID_VALUES	Especifica un conjunto de valores válidos para una columna. Utilice un punto y coma para separar varios valores.
XPATH	Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.

Información de consulta sobre opciones de Sybase

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 67. Opciones de derivador para Sybase

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. En Microsoft Windows, los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable). En UNIX, el valor predeterminado y único valor válido es Y (el derivador debe ejecutarse en modalidad delimitada).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 68. Opciones de servidor para Sybase

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.

Tabla 68. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
CONV_EMPTY_STRING	Especifica si el servidor federado convierte una serie vacía en un espacio durante las tareas de duplicación. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado no convierte series vacías). Establezca esta opción en Y si el origen de datos tiene una columna de caracteres que no admite nulos que almacena una serie vacía.
DBNAME	Obligatoria. Especifica el nombre de la base de datos a la que desea acceder. Obtenga el nombre de la base de datos del servidor Sybase.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.

Tabla 68. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_TWO_PHASE_COMMIT	Especifica si el servidor federado se conecta con el origen de datos mediante el protocolo de confirmación en dos fases o mediante el protocolo de confirmación en una fase. Los valores válidos son Y y N. El valor predeterminado es N (el servidor federado utiliza el protocolo de confirmación en una fase para conectarse). Y especifica que el servidor federado utiliza el protocolo de confirmación en dos fases para conectarse.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
FOLD_ID	Especifica si el ID de usuario se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía el ID de usuario en mayúsculas y, si éste falla, el servidor lo envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.
FOLD_PW	Especifica si la contraseña se va a enviar al origen de datos en mayúsculas o en minúsculas. No hay valor predeterminado (el servidor federado envía la contraseña en mayúsculas y, si ésta falla, el servidor la envía en minúsculas). Los valores válidos son U (mayúsculas), L (minúsculas) y N (nulo). Evite utilizar el valor nulo, ya que podría afectar al rendimiento.

Tabla 68. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
IFILE	Especifica la vía de acceso y el nombre del archivo de interfaz de Sybase que se va a utilizar en lugar del archivo de interfaz predeterminado. El derivador Sybase busca el archivo de interfaz en los lugares siguientes, en el orden especificado: en Microsoft Windows, en la opción de servidor IFILE, después en el directorio %DB2PATH%\interfaces y, finalmente, en el directorio %SYBASE%\ini\sql.ini. En UNIX, en la opción de servidor IFILE, a continuación en el directorio sqllib/interfaces y, finalmente, en el directorio \$SYBASE/interfaces.
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
LOGIN_TIMEOUT	Especifica el tiempo, en segundos, que el servidor federado esperará antes de abandonar una solicitud de inicio de sesión. El valor predeterminado es 0 (el servidor federado espera de forma indefinida).
NODE	Obligatoria. Especifica el nombre del nodo en que reside el servidor Sybase. El nombre de nodo se encuentra en el archivo de interfaces de Sybase.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.

Tabla 68. Opciones de servidor para Sybase (continuación)

Nombre	Descripción
PACKET_SIZE	Especifica el tamaño de paquete, en bytes, que utiliza la biblioteca de cliente para enviar paquetes de secuencia de datos tabular (TDS). Si el derivador Sybase necesita enviar o recibir grandes cantidades de datos de tipo texto e imagen, aumente el valor de PACKET_SIZE.
PLAN_HINTS	Especifica si se van a habilitar las sugerencias de plan o no. Las sugerencias de plan son fragmentos de sentencias que proporcionan información adicional que el optimizador de orígenes de datos utiliza para mejorar el rendimiento de la consulta. El optimizador de orígenes de datos utiliza las sugerencias de plan para decidir si utilizar un índice o no y qué índice o qué secuencia de unión de tabla utilizar. Los valores válidos son Y y N. El valor predeterminado es N (las sugerencias de plan no se habilitan). Y especifica que las sugerencias de plan se van a habilitar en el origen de datos si éste admite las sugerencias de plan.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.
TIMEOUT	Especifica el tiempo máximo, en segundos, que el servidor federado esperará una respuesta del servidor remoto a un mandato. El valor predeterminado es 0, que especifica que el tiempo de espera es indefinido.
XA_OPEN_STRING_OPTIONS	Especifica series abiertas para la interfaz Sybase DTM XA. Estas series se añaden al nombre del gestor de recursos lógicos, el nombre de usuario y la contraseña.

Opciones de correlación de usuarios

Tabla 69. Opciones de correlación de usuarios para Sybase

Opción	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 70. Opciones de columna para Sybase

Opción	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de Teradata

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 71. Opciones de derivador para Teradata

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. En Microsoft Windows, los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable). En UNIX, el valor predeterminado es Y (el valor derivador debe ejecutarse en modalidad delimitada).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Opciones de servidor

Tabla 72. Opciones de servidor para Teradata

Nombre	Descripción
COLLATING_SEQUENCE	Especifica si el origen de datos utiliza la misma secuencia de clasificación por omisión que la base de datos federada. Los valores válidos son Y, N e I. I especifica que no distingue entre mayúsculas y minúsculas. El valor predeterminado es Y. La secuencia de clasificación especificada para el servidor federado debe coincidir con la secuencia de clasificación del origen de datos remoto.
COMM_RATE	Especifica la velocidad de comunicación, en megabytes por segundo, entre el servidor federado y el servidor de orígenes de datos. Los valores válidos son los números enteros mayores que 0 y menores que 2147483648. El valor predeterminado es 2.

Tabla 72. Opciones de servidor para Teradata (continuación)

Nombre	Descripción
CPU_RATIO	Especifica lo rápida o lenta que es la CPU del servidor de orígenes de datos en comparación con la CPU del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de CPU (una relación 1:1). El valor 0,5 indica que la velocidad de CPU del servidor federado es un 50% más lenta que la CPU del servidor de orígenes de datos. El valor 2 indica que la CPU del servidor federado es el doble de rápida que la CPU del servidor de orígenes de datos.
DB2_MAXIMAL_PUSHDOWN	Especifica los principales criterios que utiliza el optimizador de consultas para seleccionar un plan de acceso. Los valores válidos son Y y N. El valor predeterminado es N (el optimizador de consultas selecciona el plan que tiene el coste estimado menor). Y especifica que el optimizador de consultas selecciona el plan de acceso que envía la mayoría de las operaciones de consulta al origen de datos.
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 72. Opciones de servidor para Teradata (continuación)

Nombre	Descripción
IO_RATIO	Especifica lo rápido o lento que es el sistema de E/S del servidor de orígenes de datos en comparación con el sistema de E/S del servidor federado. Los valores válidos son los números mayores que 0 y menores que 1×10^{23} . El valor predeterminado es 1,0. Los valores pueden expresarse en cualquier notación doble válida, como por ejemplo, 123E10, 123 ó 1,21E4. El valor 1 indica que el servidor federado y el servidor de orígenes de datos tienen la misma velocidad de E/S (una relación 1:1). El valor 0,5 indica que la velocidad del servidor federado es un 50% más lenta que la del servidor de orígenes de datos. El valor 2 indica que la velocidad del servidor federado es el doble de rápida que la del servidor de orígenes de datos.
NODE	Obligatoria. Especifica el nombre de alias o la dirección IP del servidor Teradata.
OLD_NAME_GEN	Especifica cómo convertir los nombres de columna y los nombre de índice del origen de datos en nombres de columna de apodo y nombres de índice local del servidor federado. Los valores válidos son Y y N. El valor predeterminado es N (los nombres generados son muy parecidos a los nombres del origen de datos). Y especifica que los nombres generados son los mismos que los nombres creados en IBM WebSphere Federation Server Versión 9 y versiones anteriores. Por lo tanto, puede que los nombres no coincidan exactamente con los nombres del origen de datos.
PUSHDOWN	Especifica si el servidor federado permite que el origen de datos evalúe operaciones. Los valores válidos son Y y N. El valor predeterminado es Y (el origen de datos evalúa operaciones). N especifica que el servidor federado envía sentencias de SQL que incluyen sólo SELECT con nombres de columna. En las sentencias de SQL que el servidor federado envía al origen de datos no se incluyen predicados, como WHERE=, funciones de columna y escalares, como MAX y MIN, clasificaciones, como ORDER BY o GROUP BY ni uniones.

Opciones de correlación de usuarios

Tabla 73. Opciones de correlación de usuarios para Teradata

Nombre	Descripción
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.

Opciones de columna

Tabla 74. Opciones de columna para Teradata

Nombre	Descripción
NUMERIC_STRING	Especifica cómo tratar las series numéricas. El valor predeterminado es N. Si la columna de tipo serie del origen de datos sólo contiene series numéricas y ningún otro tipo de caracteres, incluidos los blancos, establezca la opción NUMERIC_STRING en Y. Si NUMERIC_STRING se establece en Y para una columna, el optimizador de consultas reconoce que la columna no contiene blancos que puedan interferir con la clasificación de los datos de la columna. Utilice esta opción si la secuencia de clasificación de un origen de datos es distinta de la secuencia de clasificación que utiliza el servidor federado. Las columnas que utilizan esta opción no se excluyen de la evaluación remota debido a que utilizan una secuencia de clasificación diferente.

Información de consulta sobre opciones de archivos con estructura de tabla

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de apodo y de columna.

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Opciones de derivador

Tabla 75. Opciones de derivador para archivos con estructura de tabla

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).

Opciones de servidor

Tabla 76. Opciones de servidor para archivos con estructura de tabla

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Opciones de apodo

Tabla 77. Opciones de apodo para archivos con estructura de tabla

Nombre	Descripción
COLUMN_DELIMITER	Especifica un solo carácter que se utilizará como delimitador que separa las columnas del archivo con estructura de tabla. El valor predeterminado es la coma (.). Los apóstrofes no pueden utilizarse como delimitadores. El delimitador de columna debe ser el mismo en todo el archivo. Un valor nulo se representa mediante dos delimitadores juntos o por un delimitador seguido de un terminador de línea, si el campo NULL es el último de la línea. El delimitador de columna no puede existir como datos válidos de una columna.
CODEPAGE	Especifica la página de códigos del archivo del origen de datos. Esta opción sólo es válida para bases de datos federadas que utilizan Unicode. Los datos fuente se convierten a partir de la página de códigos especificada en Unicode.

Tabla 77. Opciones de apodo para archivos con estructura de tabla (continuación)

Nombre	Descripción
FILE_PATH	<p>Especifica la vía de acceso totalmente calificada del archivo con estructura de tabla. Escriba el nombre de archivo entre apóstrofos. El archivo de datos debe ser un archivo estándar o un enlace simbólico, en vez de un conducto u otro tipo de archivo que no sea estándar.</p> <p>Importante: Si especifica la opción FILE_PATH, no especifique una columna DOCUMENT.</p>
KEY_COLUMN	<p>Especifica el nombre de la columna que se utiliza para ordenar el archivo. Una columna que tenga la opción de columna DOCUMENT no puede utilizar la columna de clave. Sólo se admiten las claves de una sola columna. El valor debe ser el nombre de una columna definida en la sentencia CREATE NICKNAME. La columna debe ordenarse de forma ascendente. Debe especificarse que la columna de clave no admite nulos añadiendo la opción NOT NULL a su definición en la sentencia de apodo. Este valor es sensible a mayúsculas y minúsculas.</p>
SORTED	<p>Especifica si el archivo del origen de datos está ordenado de forma ascendente o no. Los valores válidos son Y y N. El valor predeterminado es N (el archivo del origen de datos no está ordenado de forma ascendente). Los orígenes de datos ordenados deben ordenarse de forma ascendente según la secuencia de clasificación del entorno local actual, tal como definen los valores de la categoría de soporte multilingüístico LC_COLLATE. Si especifica que el origen de datos está ordenado, establezca la opción VALIDATE_DATA_FILE en Y.</p>
VALIDATE_DATA_FILE	<p>En el caso de archivos ordenados, esta opción especifica si el derivador ha de verificar que la columna de clave está ordenada de forma ascendente y si ha de comprobar la existencia de claves nulas. Esta validación se realiza sólo una vez la primera vez que se crea el apodo. El valor predeterminado es N (no se verifica el orden de clasificación). Esta opción sólo es válida si la opción SORTED se establece en Y y si no se especifica la opción DOCUMENT.</p>

Opciones de columna

Tabla 78. Opciones de columna para archivos con estructura de tabla

Opción	Descripción
DOCUMENT	Permite especificar la vía de acceso del archivo cuando se ejecuta la consulta y no cuando se crea el apodo. El único valor válido es FILE. Con la opción DOCUMENT sólo puede especificarse una columna de cada apodo. El tipo de datos de la columna asociada con la opción DOCUMENT debe ser VARCHAR o CHAR. La utilización de la opción de columna de apodo DOCUMENT en lugar de la opción de apodo FILE_PATH implica que el archivo que corresponde a este apodo se suministra cuando se ejecuta la consulta. Si la opción DOCUMENT tiene el valor FILE, el valor que se suministra cuando se ejecuta la consulta es la vía de acceso completa del archivo cuyo esquema coincide con la definición de este apodo.

Información de consulta sobre opciones de servicios web

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 79. Opciones de derivador para servicios web

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. El valor predeterminado es N (el derivador se ejecuta en modalidad fiable).
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.

Tabla 79. Opciones de derivador para servicios web (continuación)

Nombre	Descripción
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada. El valor predeterminado es <i>vía de acceso de instalación/cfg/WSWrapperKeystore.kdb</i> .
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <i>file:nombre_archivo</i> , donde <i>nombre_archivo</i> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si el certificado del servidor se verifica durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de servidor

Tabla 80. Opciones de servidor para servicios web

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.

Tabla 80. Opciones de servidor para servicios web (continuación)

Nombre	Descripción
DB2_UM_PLUGIN	Especifica la implementación del conector de correlaciones de usuarios. En el caso de un conector escrito en Java, especifica una serie que es sensible a mayúsculas y minúsculas para el nombre de clase que corresponde a la clase de repositorio de correlación de usuarios. Por ejemplo, "UserMappingRepositoryLDAP". En el caso de un conector escrito en C, especifica un nombre válido de una biblioteca C.
DB2_UM_PLUGIN_LANG	Especifica el lenguaje del conector de correlaciones de usuarios. Los valores válidos son Java y C. El valor predeterminado es Java.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada. El valor predeterminado es <i>vía de acceso de instalación/cfg/WSWrapperKeystore.kdb</i> .

Tabla 80. Opciones de servidor para servicios web (continuación)

Nombre	Descripción
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si hay que verificar el certificado del servidor durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de correlación de usuarios

Tabla 81. Opciones de correlación de usuarios para servicios web

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy. La contraseña se cifra al almacenarla en el catálogo de la base de datos federada.
REMOTE_AUTHID	Especifica el ID de usuario remoto con que está correlacionado el ID de usuario local. Si no especifica esta opción, se usará el ID utilizado para conectarse con la base de datos federada.
REMOTE_PASSWORD	Especifica la contraseña remota del ID de usuario remoto. Si no especifica esta opción, se usará la contraseña utilizada para conectarse con la base de datos federada.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.

Opciones de apodo

Tabla 82. Opciones de apodo para servicios web

Nombre	Descripción
NAMESPACES	<p>Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis:</p> <pre>NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"'</pre> <p>Utilice comas para separar varios espacios de nombres. Por ejemplo:</p> <pre>NAMESPACES='http://www.miweb.com/clie", i='http://www.miweb.com/clie/id", n='http://www.miweb.com/clie/nombre"'</pre>
SOAPACTION	<p>Opción obligatoria para el apodo raíz. Especifica el atributo URI SOAPACTION del formato del lenguaje de descripción de servicios web (WSDL). El URL puede contener una dirección IPv6 separada por dos puntos si se escribe entre corchetes. Por ejemplo:http:// [1080:0:0:0:8:800:200C:417A]</p> <p>Nota: Esta opción no es válida para apodos distintos del apodo raíz.</p>
STREAMING	<p>Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz.</p>
TEMPLATE	<p>Especifica el fragmento de plantilla de apodo que se va a utilizar para crear una solicitud SOAP. El fragmento debe ajustarse a la sintaxis especificada para la plantilla. Esta opción sólo es válida para el apodo raíz.</p>
URL	<p>Opción obligatoria para el apodo raíz. Especifica el URL del punto final de servicio web. Los protocolos admitidos son HTTP y HTTPS. El URL puede contener una dirección IPv6 separada por dos puntos si se escribe entre corchetes. Por ejemplo:http:// [1080:0:0:0:8:800:200C:417A]</p>

Tabla 82. Opciones de apodo para servicios web (continuación)

Nombre	Descripción
XML_CODESET	Especifica la codificación que hay que utilizar para enviar y recibir datos XML. Esta opción altera temporalmente la codificación interna.
XPATH	Obligatoria. Especifica la expresión XPath que identifica los elementos de respuesta SOAP que representan tuplas individuales. La expresión XPath se utiliza como contexto para evaluar valores de columnas que identifican las opciones de columna de apodo XPATH.

Opciones de columna

Tabla 83. Opciones de columna para servicios web

Nombre	Descripción
ESCAPE_INPUT	Especifica si hay que sustituir caracteres XML especiales en los valores XML de entrada o no. Utilice esta opción para incluir fragmentos XML como entrada, por ejemplo, para incluir fragmentos XML que tienen elementos repetidos. Los valores válidos son Y y N. N es el valor predeterminado (los valores XML de entrada se mantienen). El tipo de datos de columna debe ser VARCHAR o CHAR. Si ESCAPE_INPUT se establece en Y, también deberá especificar la opción de columna TEMPLATE.

Tabla 83. Opciones de columna para servicios web (continuación)

Nombre	Descripción
FOREIGN_KEY	<p>Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.
PRIMARY_KEY	<p>Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.

Tabla 83. Opciones de columna para servicios web (continuación)

Nombre	Descripción
SOAPACTIONCOLUMN	<p>Especifica la columna que especifica dinámicamente la acción SOAP para el punto final de servicios web cuando se ejecuta una consulta. Esta opción sólo es válida para el apodo raíz. Si el nombre de host es una dirección IPv6 (separada por dos puntos), escriba el nombre de host entre corchetes. Por ejemplo: 'http://[1080:0:0:0:8:800:200C:417A]:99/soap'. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.</p>
TEMPLATE	<p>Especifica el fragmento de plantilla de columna que se va a utilizar para crear el documento XML de entrada. El fragmento debe ajustarse a la sintaxis especificada para la plantilla.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>
URLCOLUMN	<p>Especifica la columna que especifica dinámicamente la acción SOAP para el punto final de servicios web cuando se ejecuta una consulta. Esta opción sólo es válida para el apodo raíz. Si el nombre de host es una dirección IPv6 (separada por dos puntos), escriba el nombre de host entre corchetes. Por ejemplo: 'http://[1080:0:0:0:8:800:200C:417A]:99/soap'. Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna.</p>
XPATH	<p>Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>

Información de consulta sobre opciones de XML

Se explica cómo configurar la forma en que interactúan el servidor federado y sus usuarios con un origen de datos, establecer y modificar las opciones de derivador, de servidor, de correlación de usuarios, de apodo y de columna.

Opciones de derivador

En las tablas siguientes se indican las opciones que son aplicables a este origen de datos y se identifican las opciones obligatorias que deben especificarse.

Tabla 84. Opciones de derivador para XML

Nombre	Descripción
DB2_FENCED	Obligatoria. Especifica si el derivador se ejecuta en modalidad delimitada o en modalidad fiable. Los valores válidos son Y y N. N es el valor predeterminado (el derivador se ejecuta en modalidad fiable).
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada.
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si el certificado del servidor se verifica durante la autenticación SSL. Los valores válidos son Y y N. El valor predeterminado es N (el certificado no se verifica).

Opciones de servidor

Tabla 85. Opciones de servidor para XML

Nombre	Descripción
DB2_MAX_ASYNC_REQUESTS_PER_QUERY	Especifica el número máximo de solicitudes asíncronas simultáneas de una consulta. Los valores válidos son los comprendidos entre -1 y 64000. El valor predeterminado es 1. -1 especifica que el optimizador de consultas federado determina el número de solicitudes. 0 especifica que el origen de datos no puede dar cabida a solicitudes asíncronas adicionales.
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
PROXY_SERVER_NAME	Especifica el nombre o la dirección IP del servidor proxy. Las direcciones IP válidas están en formato IPv4 (separadas por puntos) o en formato IPv6 (separadas por dos puntos). Utilice el formato IPv6 únicamente si está configurado el protocolo IPv6.
PROXY_SERVER_PORT	Especifica el puerto o el nombre de servicio del servicio proxy en el servidor proxy. Los valores válidos son un número de puerto decimal comprendido entre 1 y 32760 o un nombre de servicio.
PROXY_TYPE	Especifica el tipo de proxy que hay que utilizar para acceder a Internet si el servidor federado está tras un cortafuegos. Los valores válidos son NONE, HTTP y SOCKS. El valor predeterminado es NONE.
SOCKET_TIMEOUT	Especifica el tiempo máximo, en minutos, que el servidor federado esperará algún resultado del servidor proxy. Un valor válido es cualquier número que sea mayor o igual que 0. El valor predeterminado es 0 (el servidor esperará de forma indefinida).
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.
SSL_KEYSTORE_FILE	Especifica el archivo de almacenamiento de certificados para las comunicaciones que utilizan SSL o TSL. Un valor válido es un nombre de vía de acceso totalmente calificada al que pueda acceder el agente de la base de datos federada o un proceso de modalidad delimitada.

Tabla 85. Opciones de servidor para XML (continuación)

Nombre	Descripción
SSL_KEYSTORE_PASSWORD	Especifica la contraseña que hay que utilizar para acceder al archivo de la opción SSL_KEYSTORE_FILE. Los valores válidos son una contraseña, que se cifra al almacenarla en el catálogo de la base de datos federada, y <code>file:nombre_archivo</code> , donde <code>nombre_archivo</code> es la vía de acceso totalmente calificada de un archivo stash.
SSL_VERIFY_SERVER_CERTIFICATE	Especifica si hay que verificar el certificado del servidor durante la autenticación SSL. El valor predeterminado es N (el certificado no se verifica).

Opciones de correlación de usuarios

Tabla 86. Opciones de correlación de usuarios para XML

Nombre	Descripción
PROXY_AUTHID	Especifica el nombre de usuario de la autenticación de servidor proxy.
PROXY_PASSWORD	Especifica la contraseña de la autenticación de servidor proxy.
SSL_CLIENT_CERTIFICATE_LABEL	Especifica el certificado del cliente que se ha de enviar durante la autenticación SSL. Si no especifica un valor, se utilizará el ID de autorización actual de la base de datos federada para encontrar el certificado.

Opciones de apodo

Tabla 87. Opciones de apodo para XML

Nombre	Descripción
DIRECTORY_PATH	Especifica el nombre de la vía de acceso de un directorio que contiene uno o varios archivos XML. Utilice esta opción para crear un único apodo para varios archivos de origen XML. El derivador XML utiliza solamente los archivos con extensión <code>.xml</code> que se encuentran en el directorio especificado. El derivador XML no tiene en cuenta los otros archivos del directorio. Si especifica esta opción de apodo, no especifique una columna DOCUMENT. Esta opción sólo es válida para el apodo raíz.
FILE_PATH	Especifica la vía de acceso del archivo del documento XML. Si especifica FILE_PATH, no especifique una columna DOCUMENT. Esta opción sólo es válida para el apodo raíz.

Tabla 87. Opciones de apodo para XML (continuación)

Nombre	Descripción
INSTANCE_PARSE_TIME	<p>Especifica el tiempo, en milisegundos, necesario para analizar una fila del documento fuente XML. El valor válido puede ser un entero o un valor decimal. El valor predeterminado es 7. Esta opción sólo es válida para columnas del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.</p>
NAMESPACES	<p>Especifica los espacios de nombres asociados con los prefijos de espacio de nombres que se utilizan en las opciones XPATH y TEMPLATE de cada columna. Utilice esta sintaxis:</p> <pre data-bbox="963 743 1451 827">NAMESPACES'prefijo1="espacio_de_nombres_real1", prefijo2="espacio_de_nombres_real2"'</pre> <p>Utilice comas para separar varios espacios de nombres. Por ejemplo:</p> <pre data-bbox="963 926 1451 1003">NAMESPACES='http://www.miweb.com/clie", i='http://www.miweb.com/clie/id", n='http://www.miweb.com/clie/nombre"'</pre>
NEXT_TIME	<p>Especifica el tiempo, en milisegundos, necesario para encontrar elementos fuente posteriores de la expresión XPath. El valor predeterminado es 1. Esta opción es válida para apodos raíz y distintos del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.</p>
STREAMING	<p>Especifica si el documento fuente debe dividirse en fragmentos lógicos para procesarlo. Los fragmentos corresponden al nodo que coincide con la expresión XPath del apodo. A continuación, el derivador analiza y procesa los datos del documento fuente fragmento a fragmento. Este tipo de análisis minimiza la utilización de memoria. Los valores válidos son Y y N. El valor predeterminado es N (los documentos no se analizan). Esta opción sólo es válida en el apodo raíz.</p>

Tabla 87. Opciones de apodo para XML (continuación)

Nombre	Descripción
VALIDATE	Especifica si el documento fuente se valida para asegurarse de que se ajusta a un esquema XML o a una definición de tipo de documento (DTD) antes de que los datos se extraigan de él. El valor predeterminado es N (la validación no se realiza). Antes de establecer el valor en Y, el archivo de esquema o el archivo DTD ha de estar en la ubicación especificada por el documento fuente. Esta opción sólo es válida para el apodo raíz. No establezca las opciones STREAMING y VALIDATE en Y a la vez.
XPATH	Obligatoria. Especifica la expresión XPath que identifica los elementos que representan las tuplas individual. La expresión XPath se utiliza como contexto para evaluar valores de columnas identificadas por la opción de columna XPATH.
XPATH_EVAL_TIME	Especifica el tiempo necesario, en milisegundos, para evaluar la expresión XPath del apodo y para encontrar el primer elemento. El valor puede ser un entero o un valor decimal. El valor predeterminado es 1. Esta opción es válida para apodos raíz y distintos del apodo raíz. Para optimizar consultas de estructuras fuente XML complejas o de gran tamaño, modifique las opciones INSTANCE_PARSE_TIME, XPATH_EVAL_TIME y NEXT_TIME.

Opciones de columna

Tabla 88. Opciones de columna para XML

Nombre	Descripción
DOCUMENT	Especifica que esta columna es una columna DOCUMENT. El valor de la columna DOCUMENT indica el tipo de datos fuente XML suministrado al apodo cuando se ejecuta la consulta. Esta opción sólo es válida para columnas del apodo raíz (el apodo que identifica los elementos en el nivel superior del documento XML). Con la opción DOCUMENT sólo puede especificarse una columna para cada apodo. Si utiliza una opción de columna DOCUMENT en vez de una opción de apodo FILE_PATH o DIRECTORY_PATH, el documento que corresponde a este apodo se suministra cuando se ejecuta la consulta. Los valores válidos son FILE, DIRECTORY, URI y COLUMN. El valor predeterminado es FILE. El tipo de datos de la columna debe ser VARCHAR.

Tabla 88. Opciones de columna para XML (continuación)

Nombre	Descripción
FOREIGN_KEY	<p>Indica que este apodo es un apodo hijo y especifica el nombre del apodo padre correspondiente. Un apodo puede tener como mucho una opción de columna FOREIGN_KEY. El valor de esta opción es sensible a mayúsculas y minúsculas. No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir un apodo padre y un apodo hijo. Una sentencia CREATE NICKNAME que incluya la opción FOREIGN_KEY fallará si el apodo padre tiene un nombre de esquema diferente. A menos que el apodo al que se hace referencia en la cláusula FOREIGN_KEY se hubiera definido en la sentencia CREATE NICKNAME explícitamente en minúsculas o en una combinación de mayúsculas y minúsculas, deberá especificar el apodo en mayúsculas cuando haga referencia a éste en la cláusula FOREIGN_KEY.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.
PRIMARY_KEY	<p>Opción obligatoria para un apodo padre que tiene uno o varios apodos hijo. Especifica que este apodo es un apodo padre. El tipo de datos de columna debe ser VARCHAR(16). Un apodo puede tener solamente una opción de columna PRIMARY_KEY. El único valor válido es Yes (Sí). No especifique la opción XPATH para esta columna. La columna sólo puede utilizarse para unir apodos padre y apodos hijo.</p> <p>Nota:</p> <ul style="list-style-type: none"> • Si se establece esta opción en una columna, no se podrán establecer más opciones para la columna. • Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.

Tabla 88. Opciones de columna para XML (continuación)

Nombre	Descripción
XPATH	<p>Especifica la expresión XPath en el documento XML que contiene los datos correspondientes a esta columna. El derivador evalúa esta expresión XPath después de que la sentencia CREATE NICKNAME aplique la expresión XPath de la opción de apodo XPATH.</p> <p>Nota: Si se establece esta opción de columna, luego no podrá utilizar la sentencia ALTER NICKNAME para descartar la opción. Lo que deberá hacer es descartar el apodo y, a continuación, crearlo de nuevo sin esta opción de columna.</p>

Capítulo 37. Vistas de la tabla de catálogo global que contienen información federada

La mayoría de vistas de catálogo de una base de datos federada son iguales que las vistas de catálogo de cualquier otra base de datos DB2 para Linux, UNIX y Windows.

Hay algunas vistas exclusivas que contienen información relacionada con un sistema federado, por ejemplo, la vista SYSCAT.WRAPPERS.

Las vistas SYSCAT son de sólo lectura. No puede emitir una operación de actualización o inserción en una vista del esquema SYSCAT. Las vistas SYSSTAT son la forma recomendada de actualizar el catálogo de sistema. Cambie las aplicaciones que hacen referencia a la vista SYSCAT para hacer referencia a la vista SYSSTAT actualizable.

La tabla siguiente lista las vistas SYSCAT que contienen información federada. Estas vistas son de sólo lectura.

Tabla 89. Vistas de catálogo que suele utilizarse con un sistema federado

Vistas de catálogo	Descripción
SYSCAT.CHECKS	Contiene la información de restricciones de comprobación definida por el usuario.
SYSCAT.COLCHECKS	Contiene las columnas referenciadas por una restricción de comprobación.
SYSCAT.COLUMNS	Contiene información de columna sobre los objetos de orígenes de datos (tablas y vistas) para los que ha creado apodos.
SYSCAT.COLOPTIONS	Contiene la información sobre los valores de opciones de columna que ha establecido para un apodo.
SYSCAT.CONSTDEP	Contiene la dependencia de una restricción informativa definida por el usuario.
SYSCAT.DATATYPES	Contiene información de tipo de datos sobre tipos de datos DB2 locales incorporados y definidos por el usuario.
SYSCAT.DBAUTH	Contiene las autoridades de bases de datos mantenidas por grupos y usuarios individuales.
SYSCAT.FUNCMAPOPTIONS	Contiene información sobre los valores de opciones que ha establecido para una correlación de funciones.
SYSCAT.FUNCMAPPINGS	Contiene las correlaciones de funciones entre la base de datos federada y los objetos de origen de datos.
SYSCAT.INDEXCOLUSE	Contiene las columnas que forman parte de un índice.
SYSCAT.INDEXES	Contienen las especificaciones de índice para los objetos de origen de datos.

Tabla 89. Vistas de catálogo que suele utilizarse con un sistema federado (continuación)

Vistas de catálogo	Descripción
SYSCAT.INDEXOPTIONS	Contiene la información sobre las opciones de índice.
SYSCAT.KEYCOLUSE	Contiene las columnas que forman parte de una clave definida por una restricción de clave exclusiva, de clave primaria o de clave foránea.
SYSCAT.NICKNAMES	Contiene la información sobre los apodos que ha creado.
SYSCAT.REFERENCES	Contiene la información sobre las restricciones referenciales que ha definido.
SYSCAT.ROUTINES	Contiene funciones DB2 locales definidas por el usuario, o plantillas de función. Las plantillas de función se utilizan para correlacionar a una función de origen de datos.
SYSCAT.REVTYPEMAPPINGS	Esta vista no se utiliza. Todas las correlaciones de tipos de datos se registran en la vista SYSCAT.TYPEMAPPINGS.
SYSCAT.ROUTINEOPTIONS	Contiene la información sobre los valores de opciones de rutinas federadas.
SYSCAT.ROUTINEPARMOPTIONS	Contiene la información sobre los valores de opciones de parámetros de rutinas federadas.
SYSCAT.ROUTINEPARMS	Contiene un parámetro o el resultado de una rutina definida en SYSCAT.ROUTINES.
SYSCAT.ROUTINESFEDERATED	Contiene la información sobre las rutinas federadas que ha definido.
SYSCAT.SERVERS	Contiene definiciones de servidor creadas para los servidores de orígenes de datos.
SYSCAT.TABCONST	Cada fila representa una tabla y restricciones de apodo del tipo CHECK, UNIQUE, PRIMARY KEY o FOREIGN KEY.
SYSCAT.TABLES	Contiene información sobre cada apodo, vista federada y tabla DB2 local que cree.
SYSCAT.TYPEMAPPINGS	Contiene correlaciones de tipos de datos directas y correlaciones de tipos de datos inversas. La correlación es a tipos de datos DB2 locales de tipos de datos de orígenes de datos. Estas correlaciones se utilizan cuando crea un apodo en un objeto de orígenes de datos.
SYSCAT.USEROPTIONS	Contiene la información de autorización de usuario establecida al crear correlaciones de usuarios entre la base de datos federada y los servidores de orígenes de datos.
SYSCAT.VIEWS	Contiene la información sobre las vistas federadas locales que ha creado.
SYSCAT.WRAPOPTIONS	Contiene la información sobre los valores de opciones que ha establecido para un derivador.

Tabla 89. Vistas de catálogo que suele utilizarse con un sistema federado (continuación)

Vistas de catálogo	Descripción
SYSCAT.WRAPPERS	Contiene el nombre del derivador y el archivo de biblioteca para cada origen de datos para el que cree un derivador.

La tabla siguiente lista las vistas SYSSTAT que contienen información federada. Estas son vistas de sólo lectura que contienen estadísticas que puede actualizar.

Tabla 90. Vistas de catálogo global federadas y actualizables

Vistas de catálogo	Descripción
SYSSTAT.COLUMNS	Contiene información estadística sobre cada columna de los objetos de orígenes de datos (tablas y vistas) para los que ha creado apodos. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.
SYSSTAT.INDEXES	Contienen información estadística sobre cada especificación de índice para los objetos de origen de datos.
SYSSTAT.ROUTINES	Contiene información estadística sobre cada función definida por el usuario. No incluye funciones incorporadas. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.
SYSSTAT.TABLES	Contiene información sobre cada tabla base. la información de vista, sinónimo y alias no está incluida en esta vista. En las tablas con tipos solamente se incluye en la vista la tabla raíz de una jerarquía de tablas. Las estadísticas no se graban para las columnas heredadas de las tablas con tipos.

Capítulo 38. Opciones de correlaciones de funciones para sistemas federados

El servidor federado proporciona correlaciones predeterminadas entre las funciones de DB2 y las funciones de orígenes de datos. Para la mayor parte de orígenes de datos, las correlaciones de funciones predeterminadas se encuentran en los derivadores. Para utilizar una función de orígenes de datos que el servidor federado no reconoce o para cambiar la correlación predeterminada, cree una correlación de funciones.

Cuando crea una correlación de funciones, especifica el nombre de la función de orígenes de datos y debe habilitar la función correlacionada. Luego, cuando utiliza la función correlacionada, el optimizador de consultas compara el coste de ejecutar la función en el origen de datos con el coste de ejecutar la función en el servidor federado.

Tabla 91. Opciones para las correlaciones de funciones

Nombre	Descripción
DISABLE	Habilita o inhabilita una correlación de funciones predeterminada. Los valores válidos son Y y N. El valor predeterminado es N.
REMOTE_NAME	En nombre de la función de origen de datos. El valor predeterminado es el nombre local.

Capítulo 39. Tipos de servidor válidos en las sentencias de SQL

Los tipos de servidor indican el tipo de orígenes de datos que representa la definición de servidor.

Los tipos de servidor varían en función del proveedor, el objetivo y el sistema operativo. Los valores soportados dependen del origen de datos.

Para la mayoría de orígenes de datos, debe especificar un tipo de servidor válido en la sentencia CREATE SERVER.

Tabla 92. Orígenes de datos y tipos de servidores

Origen de datos	Tipo de servidor
BioRS	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Excel	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
IBM DB2 Universal Database para Linux, UNIX y Windows	DB2/UDB
IBM DB2 Universal Database para System i y AS/400	DB2/ISERIES
IBM DB2 Universal Database para z/OS	DB2/ZOS
IBM DB2 para VM	DB2/VM
Informix	INFORMIX
JDBC	JDBC (necesario para orígenes de datos JDBC soportados por los controladores JDBC 3.0 y posterior.)
Microsoft SQL Server	MSSQLSERVER (necesario para los orígenes de datos soportados por el controlador de DataDirect Connect ODBC 4.2 (o posterior) o por el controlador de Microsoft SQL Server ODBC 3.0 (o posterior).)
ODBC	ODBC (necesario para orígenes de datos ODBC soportados por el controlador ODBC 3.x.)
OLE DB	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Oracle	ORACLE (necesario para los orígenes de datos Oracle soportados por el software cliente Oracle NET8.)
Sybase (CTLIB)	SYBASE
Archivos con estructura de tabla	No es necesario un tipo de servidor en la sentencia CREATE SERVER.
Teradata	TERADATA
Servicios web	No es necesario un tipo de servidor en la sentencia CREATE SERVER.

Tabla 92. Orígenes de datos y tipos de servidores (continuación)

Origen de datos	Tipo de servidor
XML	No es necesario un tipo de servidor en la sentencia CREATE SERVER.

Capítulo 40. Correlaciones de tipos de datos

Las correlaciones de tipos de datos para orígenes de datos relacionales incluyen correlaciones de tipos de datos directas, correlaciones de tipos de datos inversas y correlaciones de tipos de datos que son específicas de Unicode. Cada origen de datos no relacional es compatible con determinados tipos de datos.

Correlaciones de tipos de datos directas por omisión

Los dos tipos de correlaciones entre los tipos de datos de origen de datos y los tipos de datos de base de datos federada son las correlaciones de tipos de datos directas e inversas. Una correlación de tipos de datos directa es la correlación de un tipo de datos remoto con un tipo de datos local comparable.

Con la sentencia CREATE TYPE MAPPING puede alterar temporalmente la correlación de tipos de datos por omisión o crear una correlación de tipos de datos nueva.

Estas correlaciones son válidas con todas las versiones compatibles, a menos que se indique lo contrario.

Para todas las correlaciones de tipos de datos directas por omisión de un origen de datos con la base de datos federada, el esquema federado es SYSIBM.

En las tablas siguientes se muestran las correlaciones directas por omisión entre tipos de datos de base de datos federada y tipos de datos de origen de datos.

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows.

Tabla 93. Correlaciones de tipos de datos directas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	S	-	CHAR	-	0	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DECFLOAT ²	-	-	-	-	-	-	DECFLOAT	-	0	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-

Tabla 93. Correlaciones de tipos de datos directas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	S	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP(p)	-		p	p	-	-	TIMESTAMP(p)	-	p	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	0	S
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. La opción de servidor SAME_DECFLT_ROUNDING se establece en N por omisión y las operaciones no se enviarán al origen de datos remoto a menos que SAME_DECFLT_ROUNDING se establezca en Y. Hallará más información sobre la opción de servidor SAME_DECFLT_ROUNDING en Información de consulta sobre opciones de DB2 Database.

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para System i.

Tabla 94. Correlaciones de tipos de datos directas por omisión de DB2 para System i (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
CHAR	255	32672	-	-	S	-	VARCHAR	-	0	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-

Tabla 94. Correlaciones de tipos de datos directas por omisión de DB2 para System i (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	S	-	VARCHAR	-	0	S
VARG	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para VM y VSE

Tabla 95. Correlaciones de tipos de datos directas por omisión de DB2 Server para VM y VSE (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
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	-	-	S	-	VARCHAR	-	0	S
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPH	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos DB2 para z/OS

Tabla 96. Correlaciones de tipos de datos directas por omisión de DB2 para z/OS (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	S	-	CHAR	-	0	S
CHAR	255	32672	-	-	S	-	VARCHAR	-	0	S
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	-	-	-	-	S	-	VARCHAR	40	-	S
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP(6)	-	6	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	S	-	VARCHAR	-	0	S
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

Correlaciones de tipos de datos directas por omisión para orígenes de datos Informix

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Informix.

Tabla 97. Correlaciones de tipos de datos directas por omisión de Informix (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
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	-

Tabla 97. Correlaciones de tipos de datos directas por omisión de Informix (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	-	-	-

Notas:

1. El tipo federado es `TIMESTAMP(0)` si el parámetro de configuración `date_compat` se establece en `ON`.
2. Para el tipo de datos `DATETIME` de Informix, el servidor federado `DB2 UNIX` y `Windows` utiliza el calificador de alto nivel de Informix como `REMOTE_LENGTH` (longitud remota) y el calificador de bajo nivel de Informix como `REMOTE_SCALE` (escala remota).

Los calificadores de Informix son las constantes "TU_" definidas en el archivo `datatime.h` de Informix Client SDK. Las constantes son:

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)	

Correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC.

Tabla 98. Correlaciones de tipos de datos directas por omisión de JDBC

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
BINARY	-	254	-	-	-	-	CHAR	-	-	S

Tabla 98. Correlaciones de tipos de datos directas por omisión de JDBC (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BINARY	255	32672	-	-	-	-	VARCHAR	-	-	S
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	-	-	S
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	-	-
VARBINARY	-	32672	-	-	-	-	VARCHAR	-	-	S
VARBINARY	32673	2147483647	-	-	-	-	BLOB	2147483647	-	-
VARCHAR	-	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	2147483647	-	-	-	-	CLOB	2147483647	-	-

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. Tipos de datos que sólo admite el controlador JDBC 4.0: NCHAR, NVARCHAR, LONGVARCHAR y NCLOB.

El derivador JDBC no admite los tipos de datos siguientes: DATALINK, OTHER, JAVA_OBJECT, DISTINCT, STRUCT, ARRAY y REF.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server.

Tabla 99. Correlaciones de tipos de datos directas por omisión de Microsoft SQL Server

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
bigint ¹	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	S
binary	255	8000	-	-	-	-	VARCHAR	-	-	S
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	-	S
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	-	S
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP(6)	10	6	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	S
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	S
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	-	-	S

Tabla 99. Correlaciones de tipos de datos directas por omisión de Microsoft SQL Server (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	S
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	-	-	S
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	-	S
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier	1	4000	-	-	S	-	VARCHAR	16	-	S
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	S
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

Nota:

1. Esta correlación de tipos de datos sólo es válida con Microsoft SQL Server Versión 2000.

Correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC.

Tabla 100. Correlaciones de tipos de datos directas por omisión de ODBC (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	S
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	S
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DATE	-	-	-	-	-	-	DATE	-	-	-
SQL_DATE	-	-	-	-	-	-	TIMESTAMP ¹	-	-	-

Tabla 100. Correlaciones de tipos de datos directas por omisión de ODBC (no se muestran todas las columnas) (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	-	S
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	-	-	S
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

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Oracle NET8.

Tabla 101. Correlaciones de tipos de datos directas por omisión de Oracle NET8

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	0	0	0	0	-	\0	BLOB	2147483647	0	S
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

Tabla 101. Correlaciones de tipos de datos directas por omisión de Oracle NET8 (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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	S
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	S
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

Nota:

1.

- TIMESTAMP(p) representa una indicación de fecha y hora con una escala variable que va de 0 a 9. La escala de la indicación de fecha y hora de Oracle está correlacionada con el tipo de datos TIMESTAMP(6) predeterminado. Puede cambiar la correlación de tipos de datos por omisión y correlacionar el tipo de datos TIMESTAMP de Oracle con un tipo de datos TIMESTAMP federado de la misma escala utilizando una correlación de tipos de datos definidos por el usuario.
- Esta correlación de tipos de datos sólo es válida para configuraciones de cliente y servidor Oracle 9i (o posterior).

Correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase.

Tabla 102. Correlaciones de tipos de datos directas por omisión de Sybase CTLIB

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
binary	1	254	-	-	-	-	CHAR	-	-	S
binary	255	32672	-	-	-	-	VARCHAR	-	-	S
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	32672	-	-	-	-	VARCHAR	-	-	N
char null (véase 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	-	-	-

Tabla 102. Correlaciones de tipos de datos directas por omisión de Sybase CTLIB (continuación)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
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 (véase 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	-	S
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ²	1	254	-	-	-	-	CHAR	-	-	N
unichar ²	255	32672	-	-	-	-	VARCHAR	-	-	N
unichar null (véase univarchar)										
univarchar ²	1	32672	-	-	-	-	VARCHAR	-	-	N
varbinary	1	32672	-	-	-	-	VARCHAR	-	-	S
varchar	1	32672	-	-	-	-	VARCHAR	-	-	N

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.
2. Válido para bases de datos federadas que no son Unicode.

Correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Teradata.

Tabla 103. Correlaciones de tipos de datos directas por omisión de Teradata (no se muestran todas las columnas)

Nombre de tipo remoto	Longitud inferior remota	Longitud superior remota	Escala inferior remota	Escala superior remota	Datos de bits remotos	Operadores de datos remotos	Nombre de tipo federado	Longitud federada	Escala federada	Datos de bits federados
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
BYTE	1	254	-	-	-	-	CHAR	-	-	S
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	S
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	-	-	-
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	-	-	S
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

Nota:

1. El tipo federado es TIMESTAMP(0) si el parámetro de configuración date_compat se establece en ON.

Correlaciones de tipos de datos directas de ejemplo

Puede utilizar las correlaciones de tipos de datos directas de ejemplo para sacar partido del soporte del tipo de datos TIMESTAMP con la precisión.

Para orígenes de datos Informix, estas correlaciones de tipos de datos se utilizan para tipos de columna de apodo, parámetros de procedimientos federados, paso a través y conjuntos de resultados de procedimientos federados.

Para orígenes de datos distintos de Informix, estas correlaciones de tipos de datos solamente afectan a las correlaciones para tipos de columna de apodo y parámetros de procedimientos federados. Las correlaciones no afectan a los conjuntos de resultados a través de y de procedimientos federados.

Correlaciones de tipos de datos directas - ejemplo de Informix

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Informix.

Es necesario crear estas correlaciones antes de crear un objeto federado.

```
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);
```

Correlaciones de tipos de datos directas - ejemplo de Microsoft SQL Server

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Microsoft SQL Server.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```
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";
```

Correlaciones de tipos de datos directas - ejemplo de Oracle

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Oracle.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```
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;

```

Correlaciones de tipos de datos directas - ejemplo de Sybase

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Sybase.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```

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);

```

Correlaciones de tipos de datos directas - ejemplo de Teradata

Al crear objetos federados, se puede usar la correlación de tipos de datos directa de ejemplo proporcionada por Teradata.

Es necesario crear estas correlaciones antes de crear un apodo o procedimiento federado.

```

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);

```

Correlaciones de tipos de datos inversas por omisión

Para la mayor parte de orígenes de datos, las correlaciones de tipos por omisión se encuentran en los derivadores.

Los dos tipos de correlaciones entre los tipos de datos de origen de datos y los tipos de datos de base de datos federada son las correlaciones de tipos de datos directas e inversas. Una correlación de tipos de datos directa es la correlación de un tipo de datos remoto con un tipo de datos local comparable. El otro tipo de

correlación es una correlación de tipos de datos inversa, que se utiliza con DDL transparente para crear o modificar tablas remotas.

Las correlaciones de tipos de datos por omisión para orígenes de datos de la familia DB2 están en el derivador DRDA. Las correlaciones de tipos por omisión para Informix están en el derivador INFORMIX, etc.

Cuando se define una tabla o una vista remota para la base de datos federada, la definición incluye una correlación de tipos de datos inversa. La correlación es entre un tipo de datos de base de datos federada local para cada columna y el tipo de datos remoto correspondiente. Por ejemplo, existe una correlación de tipo de datos inversa por omisión en que el tipo de datos local REAL indica el tipo de datos SMALLFLOAT de Informix.

Las bases de datos federadas no admiten los tipos de datos LONG VARCHAR, LONG VARGRAPHIC ni los definidos por el usuario.

Cuando utilice la sentencia CREATE TABLE para crear una tabla remota, especifique los tipos de datos locales que desea incluir en la tabla remota. Estas correlaciones de tipos de datos inversas por omisión asignarán los tipos remotos correspondientes a estas columnas. Por ejemplo, suponga que utiliza la sentencia CREATE TABLE para definir una tabla Informix con una columna C2 y que en la sentencia especifica BIGINT como el tipo de datos de C2. La correlación de tipos de datos inversa por omisión de BIGINT depende de la versión de Informix con que se está creando la tabla. La correlación para C2 en la tabla Informix será con DECIMAL en Informix Versión 8 y con INT8 en Informix Versión 9.

Con la sentencia CREATE TYPE MAPPING puede alterar temporalmente la correlación de tipos de datos inversa por omisión o crear una correlación de tipos de datos inversa nueva.

En las tablas siguientes se muestran las correlaciones inversas por omisión entre tipos de datos locales de base de datos federada y tipos de datos de origen de datos remoto.

Estas correlaciones son válidas con todas las versiones compatibles, a menos que se indique lo contrario.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 Database para Linux, UNIX y Windows.

Tabla 104. Correlaciones de tipos de datos inversas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits federados
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE ¹	-	4	-	-	-	-	DATE	-	-	-

Tabla 104. Correlaciones de tipos de datos inversas por omisión de DB2 Database para Linux, UNIX y Windows (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits federados
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(p)-	-	-	p	p	-	-	TIMESTAMP(p)-	-	p ³	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Nota:

1. Si el parámetro date_compat se establece en OFF, el tipo de datos DATE federado se correlaciona con el tipo de datos TIMESTAMP(0).
2. La opción de servidor SAME_DECFLT_ROUNDING se establece en N por omisión y las operaciones no se enviarán al origen de datos remoto a menos que SAME_DECFLT_ROUNDING se establezca en Y. Hallará más información sobre la opción de servidor SAME_DECFLT_ROUNDING en Información de consulta sobre opciones de DB2 Database.
3. Para la versión 9.5 (o anterior), la escala remota de TIMESTAMP es 6.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para System i.

Tabla 105. Correlaciones de tipos de datos inversas por omisión de DB2 para System i (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operaciones de datos federadas	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	S	-	CHARACTER	-	-	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-

Tabla 105. Correlaciones de tipos de datos inversas por omisión de DB2 para System i (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operaciones de datos federadas	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para VM y VSE

Tabla 106. Correlaciones de tipos de datos inversas por omisión de DB2 para VM y VSE (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP(p)	-	-	p	p	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos DB2 para z/OS

Tabla 107. Correlaciones de tipos de datos inversas por omisión de DB2 para z/OS (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	S	-	CHAR	-	-	S
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	-	-	-	-	S	-	VARCHAR	-	-	S
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Informix.

Tabla 108. Correlaciones de tipos de datos inversas por omisión de Informix

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-

Tabla 108. Correlaciones de tipos de datos inversas por omisión de Informix (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARCHAR ¹	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	S	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

Nota:

1. Esta correlación de tipos de datos sólo es válida con el servidor Informix Versión 8 (o anterior).
2. Esta correlación de tipos de datos sólo es válida con el servidor Informix Versión 9 (o posterior).

Para el tipo de datos DATETIME de Informix, el servidor federado utiliza el calificador de alto nivel de Informix como REMOTE_LENGTH (longitud remota) y el calificador de bajo nivel de Informix como REMOTE_SCALE (escala remota).

Los calificadores de Informix son las constantes "TU_" definidas en el archivo `datatime.h` de Informix Client SDK. Las constantes son:

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)	

Correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC que se ajustan a las correlaciones de tipos de datos del controlador JDBC de DB2.

Tabla 109. Correlaciones de tipos de datos inversas por omisión de JDBC

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	S	-	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	-	-	-	-	S	-	VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	VARCHAR	-	-	-

Tabla 109. Correlaciones de tipos de datos inversas por omisión de JDBC (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARGRAPHIC	-	-	-	-	-	-	NVARCHAR	-	-	-

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server.

Tabla 110. Correlaciones de tipos de datos inversas por omisión de Microsoft SQL Server (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-
VARCHAR	1	8000	-	-	S	-	varbinary	-	-	-
VARCHAR	8001	32672	-	-	S	-	image	-	-	-

Nota:

1. Esta correlación de tipos de datos sólo es válida con Microsoft SQL Server Versión 2000.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC.

Tabla 111. Correlaciones de tipos de datos inversas por omisión de ODBC (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	SQL_BIGINT	-	-	-
BLOB	-	-	-	-	-	-	SQL_LONGVARBINARY	-	-	-
CHAR	-	-	-	-	S	-	SQL_BINARY	-	-	-
CHAR	-	-	-	-	N	-	SQL_CHAR	-	-	-

Tabla 111. Correlaciones de tipos de datos inversas por omisión de ODBC (no se muestran todas las columnas) (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
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	-	-	-	-	S	-	SQL_VARBINARY	-	-	-
VARCHAR	-	-	-	-	N	-	SQL_VARCHAR	-	-	-
VARGRAPHIC	-	-	-	-	S	-	SQL_WVARCHAR	-	-	-

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Oracle NET8.

Tabla 112. Correlaciones de tipos de datos inversas por omisión de Oracle NET8

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	0	8	0	0	N	\0	NUMBER	19	0	N
BLOB	0	2147483647	0	0	S	\0	BLOB	0	0	S
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	S	\0	RAW	0	0	S
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

Tabla 112. Correlaciones de tipos de datos inversas por omisión de Oracle NET8 (continuación)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
VARCHAR	1	2000	0	0	S	\0	RAW	0	0	S

Nota:

- Si el parámetro date_compat se establece en OFF, el tipo de datos DATE federado se correlaciona con el tipo de datos de tipo fecha de Oracle. Si el parámetro date_compat se establece en ON, el tipo de datos DATE federado (equivalente al tipo de datos TIMESTAMP(0)), se correlaciona con el tipo de datos TIMESTAMP(0) de Oracle.
- Esta correlación de tipos de datos sólo es válida con Oracle Versión 8.
- TIMESTAMP(p) representa una indicación de fecha y hora con una escala variable que va de 0 a 9 para Oracle y de 0 a 12 para la federación. Si la escala es de 0 a 9, el tipo de datos TIMESTAMP remoto de Oracle tiene la misma escala que el tipo de datos TIMESTAMP federado. Si la escala del tipo de datos TIMESTAMP federado es mayor que 9, la escala correspondiente del tipo de datos TIMESTAMP de Oracle es 9, que es la mayor escala posible de Oracle.
 - Esta correlación de tipos de datos sólo es válida con Oracle Versión 9, 10 y 11.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase.

Tabla 113. Correlaciones de tipos de datos inversas por omisión de Sybase CTLIB

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	S	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
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	-	-	S	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	S	-	image	-	-	-
VARCHAR ²	1	16384	-	-	S	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	S	-	image	-	-	-

Nota:

- Esta correlación de tipos de datos sólo es válida para CTLIB con la versión 12.0 (o anterior) del servidor Sybase.
- Esta correlación de tipos de datos sólo es válida para CTLIB con la versión 12.5 (o posterior) del servidor Sybase.

Correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Teradata.

Tabla 114. Correlaciones de tipos de datos inversas por omisión de Teradata (no se muestran todas las columnas)

Nombre de tipo federado	Longitud inferior federada	Longitud superior federada	Escala inferior federada	Escala superior federada	Datos de bits federados	Operadores de datos federados	Nombre de tipo remoto	Longitud remota	Escala remota	Datos de bits remotos
BLOB	1	2097088000	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	S	-	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	-	-	-
TIME	-	-	-	-	-	-	TIME	15	-	-
TIMESTAMP	10	10	6	6	-	-	TIMESTAMP	26	6	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	S	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Nota:

1. El tipo de datos VARGRAPHIC de Teradata sólo puede contener la longitud especificada (de 1 a 32000) de un tipo de datos DBCLOB.

Correlaciones de tipos de datos por omisión Unicode

Determinados orígenes de datos son compatibles con correlaciones de tipos de datos directas y correlaciones de tipos de datos inversas para bases de datos Unicode.

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos JDBC cuando la base de datos federada es una base de datos Unicode.

Tabla 115. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC

UTF-8	JDBC
Tipo de datos	Tipo de datos Longitud
CHAR	CHAR de 1 a 254 bytes
VARCHAR	VARCHAR de 1 a 32672 bytes

Tabla 115. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos JDBC (continuación)

UTF-8	JDBC	
Tipo de datos	Tipo de datos	Longitud
CLOB	CLOB	-
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NVARCHAR	de 1 a 16336 caracteres
DBCLOB	NCLOB	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos JDBC cuando la base de datos federada es una base de datos Unicode.

Tabla 116. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos JDBC

UTF-8	JDBC	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 32672 bytes	VARCHAR
CLOB	de 1 a 2147483647 bytes	CLOB
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 16336 caracteres	NVARCHAR
DBCLOB	de 1 a 1073741823 caracteres	NCLOB

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Microsoft SQL Server cuando la base de datos federada es una base de datos Unicode.

Tabla 117. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Microsoft SQL Server

UTF-8	Microsoft SQL Server	
Tipo de datos	Tipo de datos	Longitud
CHAR	CHAR	de 1 a 254 bytes
VARCHAR	CHAR	de 255 a 8000 bytes
	VARCHAR	de 1 a 8000 bytes
CLOB	TEXT	-
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NCHAR	de 128 a 16336 caracteres
	NVARCHAR	de 1 a 16336 caracteres
DBCLOB	NTEXT	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Microsoft SQL Server cuando la base de datos federada es una base de datos Unicode.

Tabla 118. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Microsoft SQL Server

UTF-8		Microsoft SQL Server
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 32672 bytes	VARCHAR
CLOB	de 1 a 2 147 483 647 bytes	TEXT
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 16336 caracteres	NVARCHAR
DBCLOB	de 1 a 1 073 741 823 caracteres	NTEXT

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos NET8 cuando la base de datos federada es una base de datos Unicode.

Tabla 119. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos NET8

UTF-8	Oracle	
Tipo de datos	Tipo de datos	Longitud
CHAR	CHAR	de 1 a 254 bytes
VARCHAR	CHAR	de 255 a 2000 bytes
	VARCHAR2	de 1 a 4000 bytes
DBCLOB	NCLOB	
GRAPHIC	NCHAR	de 1 a 127 caracteres
VARGRAPHIC	NCHAR	de 128 a 1000 caracteres
	NVARCHAR2	de 1 a 2000 caracteres

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos NET8 cuando la base de datos federada es una base de datos Unicode.

Tabla 120. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos NET8

UTF-8		Oracle
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	CHAR
VARCHAR	de 1 a 4000 bytes	VARCHAR2
CLOB	de 1 a 2 147 483 647 bytes	CLOB
GRAPHIC	de 1 a 127 caracteres	NCHAR
VARGRAPHIC	de 1 a 2000 caracteres	NVARCHAR2
DBCLOB	de 1 a 1 073 741 823 caracteres	NCLOB

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos ODBC cuando la base de datos federada es una base de datos Unicode.

Tabla 121. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos ODBC

UTF-8	ODBC	
Tipo de datos	Tipo de datos	Longitud
CHAR	SQL_CHAR	de 1 a 254 bytes
VARCHAR	SQL_CHAR	de 255 a 32672 bytes
	SQL_VARCHAR	de 1 a 32672 bytes
CLOB	SQL_LONGVARCHAR	-
GRAPHIC	SQL_WCHAR	de 1 a 127 caracteres
VARGRAPHIC	SQL_WCHAR	de 128 a 16336 caracteres
	SQL_WVARCHAR	de 1 a 16336 caracteres
DBCLOB	SQL_WLONGVARCHAR	-

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos ODBC cuando la base de datos federada es una base de datos Unicode.

Tabla 122. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC

UTF-8	ODBC	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	SQL_CHAR
VARCHAR	de 1 a 32672 bytes	SQL_VARCHAR
CLOB	de 1 a 2 147 483 647 bytes	SQL_LONGVARCHAR
GRAPHIC	de 1 a 127 caracteres	SQL_WCHAR

Tabla 122. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos ODBC (continuación)

UTF-8		ODBC
Tipo de datos	Longitud	Tipo de datos
VARGRAPHIC	de 1 a 16336 caracteres	SQL_WVARCHAR
DBCLOB	de 1 a 1 073 741 823 caracteres	SQL_WLONGVARCHAR

Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos directas por omisión para orígenes de datos Sybase CTLIB cuando la base de datos federada es una base de datos Unicode.

Tabla 123. Correlaciones de tipos de datos directas por omisión Unicode para orígenes de datos Sybase CTLIB

UTF-8	Sybase	
Tipo de datos	Tipo de datos	Longitud
CHAR	char	de 1 a 254 bytes
	nchar	de 1 a 127 caracteres
VARCHAR	char	de 255 a 32672 bytes
	varchar	de 1 a 32672 bytes
	nchar	de 128 a 16336 caracteres
	nvarchar	de 1 a 16336 caracteres
CLOB	text	
GRAPHIC	unichar	de 1 a 127 caracteres
VARGRAPHIC	unichar	de 128 a 16336 caracteres
	univarchar	de 1 a 16336 caracteres

Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase

En la tabla siguiente se indican las correlaciones de tipos de datos inversas por omisión para orígenes de datos Sybase CTLIB cuando la base de datos federada es una base de datos Unicode.

Tabla 124. Correlaciones de tipos de datos inversas por omisión Unicode para orígenes de datos Sybase CTLIB

UTF-8	Sybase	
Tipo de datos	Longitud	Tipo de datos
CHAR	de 1 a 254 bytes	char
VARCHAR	de 1 a 32672 bytes	varchar
CLOB	de 1 a 2 147 483 647 bytes	text
GRAPHIC	de 1 a 127 caracteres	unichar
VARGRAPHIC	de 1 a 16336 caracteres	univarchar

Tipos de datos admitidos para orígenes de datos no relacionales

Para la mayoría de los orígenes de datos no relacionales, debe especificar la información de columna, incluido el tipo de datos, cuando se crean los apodos para acceder al origen de datos.

Algunos de los derivadores no relacionales crean todas las columnas necesarias para acceder a un origen de datos. Estas se conocen como *columnas fijas*. Otros derivadores permiten especificar algunos o todos los tipos de datos de las columnas en la sentencia CREATE NICKNAME.

En las secciones siguientes se indican los derivadores para los que se pueden especificar tipos de datos, y los tipos de datos que admite cada derivador.

Tipos de datos admitidos por el derivador BioRS

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador BioRS.

Tabla 125. Tipos de datos BioRS que se correlacionan con tipos de datos DB2

Tipos de datos BioRS	Tipo de datos DB2
AUTHOR	CHARACTER, CLOB, VARCHAR
DATE	CHARACTER, CLOB, VARCHAR
NUMBER	CHARACTER, CLOB, VARCHAR
REFERENCE	CHARACTER, CLOB, VARCHAR
TEXT	CHARACTER, CLOB, VARCHAR

La longitud máxima permitida para el tipo de datos CLOB es de 5 megabytes.

Tipos de datos admitidos por el derivador Excel

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador Excel.

Tabla 126. Tipos de datos Excel que se correlacionan con tipos de datos DB2

Tipos de datos Excel	Tipo de datos DB2
carácter	CHARACTER
fecha	DATE
número	DOUBLE
número	FLOAT
entero	INTEGER
carácter	VARCHAR

Tipos de datos admitidos por el derivador Script

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador Script.

Tabla 127. Tipos de datos Script que se correlacionan con tipos de datos DB2

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA

Tipos de datos admitidos por el derivador de archivos con estructura de tabla

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador de archivos con estructura de tabla

Tabla 128. Tipos de datos archivo con estructura de tabla que se correlacionan con tipos de datos DB2

Tipos de datos archivo con estructura de tabla	Tipo de datos DB2
carácter	CHARACTER
carácter	CLOB (la longitud máxima es de 5 megabytes)
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR

Tipos de datos admitidos por el derivador de servicios web

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador de servicios web. El derivador de servicios web utiliza tipos de datos XML.

Tabla 129. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador de servicios web

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER

Tabla 129. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador de servicios web (continuación)

Tipos de datos XML	Tipo de datos DB2
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA

Tipos de datos admitidos por el derivador XML

En la tabla siguiente se indican los tipos de datos DB2 que admite el derivador XML.

Tabla 130. Tipos de datos XML que se correlacionan con tipos de datos DB2 para el derivador XML

Tipos de datos XML	Tipo de datos DB2
carácter	BLOB
carácter	CHARACTER
carácter	CHARACTER FOR BIT DATA
carácter	CLOB (la longitud máxima es de 5 megabytes)
fecha	DATE
número	DECIMAL
número	DOUBLE
número	FLOAT
entero	INTEGER
número	REAL
entero	SMALLINT
carácter	VARCHAR
carácter	VARCHAR FOR BIT DATA
documento XML	XML

Documentación del producto

La documentación se proporciona en diversas ubicaciones y formatos, también en la ayuda que se abre directamente desde la interfaz del producto, en un Information Center (centro de información) para toda la suite y en manuales en archivos PDF.

También puede realizar pedidos de publicaciones de IBM en formato de copia en papel en línea o por medio de su representante local de IBM.

Para solicitar publicaciones en línea, vaya al Centro de publicaciones de IBM en www.ibm.com/shop/publications/order.

Puede enviar sus comentarios sobre la documentación de los modos siguientes:

- Formulario de comentarios del lector en línea: www.ibm.com/software/data/rcf/
- Correo electrónico: comments@us.ibm.com

Cómo leer los diagramas de sintaxis

Las reglas siguientes se aplican a los diagramas de sintaxis que se utilizan en esta documentación:

- Lea los diagramas de sintaxis de izquierda a derecha y de arriba abajo, siguiendo el recorrido de la línea. Se utilizan los convenios siguientes:
 - El símbolo >>--- indica el inicio de un diagrama de sintaxis.
 - El símbolo ---> indica que el diagrama de sintaxis continúa en la línea siguiente.
 - El símbolo >--- indica que el diagrama de sintaxis viene de la línea anterior.
 - El símbolo --->< indica el final de un diagrama de sintaxis.
- Los elementos necesarios aparecen en la línea horizontal (la línea principal).

▶▶—*elemento_necesario*—————▶▶

- Los elementos opcionales aparecen debajo de la línea principal.

▶▶—*elemento_necesario*—┐—————▶▶
 └*elemento_opcional*┘

Si aparece un elemento opcional sobre la línea principal, dicho elemento no tendrá efecto sobre el elemento de sintaxis y sólo se utilizará para facilitar la lectura.

▶▶—*elemento_necesario*—┐—————▶▶
 └*elemento_opcional*┘

- Si se puede elegir entre dos o más elementos, éstos aparecerán apilados verticalmente.

Si se debe elegir uno de los elementos, un elemento de la pila aparece en la línea principal.

▶▶—*elemento_necesario*—┐—————▶▶
 └*opción_necesaria1*┘
 └*opción_necesaria2*┘

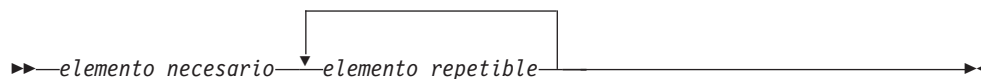
Si la elección de uno de los elementos es opcional, toda la pila aparecerá por debajo de la línea principal.

▶▶—*elemento_necesario*—┐—————▶▶
 └*opción_opcional1*┘
 └*opción_opcional2*┘

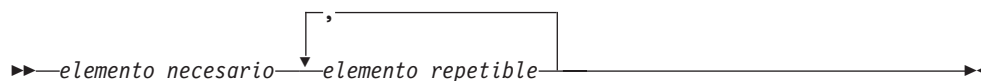
Si uno de los elementos es el predeterminado, aparecerá por encima de la línea principal y las opciones restantes se mostrarán por debajo.

▶▶—*elemento_necesario*—┐—————▶▶
 └*opción_predeterminada*┘
 └*opción_opcional1*┘
 └*opción_opcional2*┘

- Una flecha que vuelve hacia la izquierda, sobre la línea principal, indica un elemento que se puede repetir.

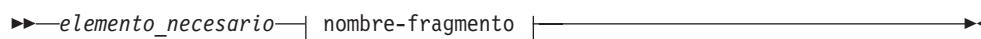


Si la flecha de repetición contiene una coma, los elementos repetidos se deben separar mediante una coma.

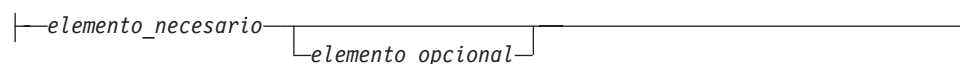


Una flecha de repetición sobre una pila indica que los elementos de la pila se pueden repetir.

- A veces, un diagrama se debe dividir en fragmentos. El fragmento de sintaxis se muestra por separado del diagrama de sintaxis principal, pero el contenido del fragmento se debe leer como si formara parte de la línea principal del diagrama.



Nombre-fragmento:



- Las palabras clave, y sus abreviaturas mínimas si las hay, aparecen en mayúsculas. Se deben escribir exactamente tal como se muestran.
- Las variables aparecen en letras minúsculas en cursiva (por ejemplo, nombre-columna). Representan nombres o valores proporcionados por el usuario.
- Separe las palabras clave y los parámetros con un espacio como mínimo si no se muestra ningún signo de puntuación en el diagrama.
- Entre los signos de puntuación, paréntesis, operadores aritméticos y otros símbolos exactamente como se muestran en el diagrama.
- Las notas a pie de página se muestran mediante un número entre paréntesis, por ejemplo (1).

Documentación accesible

La documentación se proporciona en formato XHTML, que se puede visualizar en la mayoría de los navegadores Web.

XHTML permite visualizar la documentación según las preferencias de visualización que ha establecido en el navegador. También permite utilizar lectores de pantalla y otras tecnologías de asistencia.

Los diagramas de sintaxis se proporcionan en formato decimal de puntos. Este formato solo está disponible si accede a la documentación en línea utilizando el lector de pantalla.

Avisos

La presente información se ha desarrollado para productos y servicios ofrecidos en Estados Unidos.

Es posible que IBM no comercialice en otros países los productos, servicios o características que se describen en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no otorga ninguna licencia sobre dichas patentes. Puede realizar consultas sobre licencias escribiendo a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 EE. UU.

Para formular consultas relacionadas con el juego de caracteres de doble byte (DBCS), póngase en contacto con el departamento de la propiedad intelectual de IBM de su país o envíe las consultas, por escrito, a la siguiente dirección:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japón

El párrafo siguiente no es aplicable al Reino Unido ni a ningún país en donde tales disposiciones sean incompatibles con la legislación local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN TAL CUAL, SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios Web. La información contenida en estos sitios Web no forma parte de la información del presente producto IBM, y el usuario es responsable de la utilización de dichos sitios.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
J46A/G4
555 Bailey Avenue
San José, CA 95141-1003 EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Esta información sólo tiene como objeto la planificación. La información incluida en este documento puede cambiar antes de que los productos descritos se hagan disponibles.

Este manual contiene ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos

nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

LICENCIA DE COPYRIGHT:

Este manual contiene programas de aplicaciones de ejemplo escritos en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas. Los programas de ejemplo se proporcionan "tal como están", sin garantías de ningún tipo. IBM no se hace responsable de los daños que se hayan podido causar debido al uso de los programas de ejemplo.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir una nota de copyright como la siguiente:

© (nombre de la empresa) (año). Partes de este código proceden de programas de ejemplo de IBM Corp. © Copyright IBM Corp. _entrar el año o los años_. Reservados todos los derechos.

Si examina esta información mediante una copia software, es posible que las fotografías y las ilustraciones en color no aparezcan.

Marcas registradas

Las marcas registradas de IBM y algunas marcas registradas que no son de IBM están marcadas la primera vez que aparecen en esta documentación con el símbolo adecuado.

IBM, el logotipo de IBM e ibm.com son marcas registradas de International Business Machines Corp., registradas en muchas jurisdicciones de todo el mundo. Otros nombres de productos y servicios pueden ser marcas registradas de IBM u otras empresas. Hay una lista actual de marcas registradas de IBM disponible en Internet en "Copyright and trademark information" en www.ibm.com/legal/copytrade.shtml.

Los términos siguientes son marcas registradas de otras compañías:

Adobe, el logotipo de Adobe, PostScript y el logotipo de PostScript son marcas registradas o marcas comerciales de Adobe Systems Incorporated en Estados Unidos y/o en otros países.

IT Infrastructure Library es una marca registrada de la Agencia Central de Informática y Telecomunicaciones, que es ahora parte de la Oficina de Comercio Gubernamental.

Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Intel Centrino, el logotipo de Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, y Pentium son marcas registradas de Intel Corporation y sus filiales en los Estados Unidos y otros países.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/o otros países.

Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos y/o otros países.

ITIL es una marca registrada y una marca comunitaria registrada de la Oficina de Comercio Gubernamental, y está registrada en los Estados Unidos. Oficina de patentes y marcas.

UNIX es una marca registrada de The Open Group en los Estados Unidos y en otros países.

Cell Broadband Engine es una marca registrada de Sony Computer Entertainment, Inc. en los Estados Unidos y/o otros países y está sujeta a licencia para su uso.

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos o en otros países

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de terceros.

Índice

A

accesibilidad 461, 465
acceso a datos
 vistas federadas 165
acceso a datos mediante apodos 158
actualización múltiple federada
 véase confirmación federada en dos
 fases 117
actualizaciones
 autorizaciones 143
 descripción 114
 en objetos grandes (LOB) 145
 integridad referencial 144
 locales 114
 remotas 114
 restricciones 144
actualizaciones locales 114
actualizaciones remotas 114
aislamiento a nivel de conexión
 orígenes de datos soportados 347
aislamiento a nivel de sentencia
 orígenes de datos soportados 347
aislamiento de nivel de conexión
 sistemas federados 215
aislamiento de nivel de sentencia
 orígenes de datos soportados 213
 sistemas federados 214
ajuste
 confirmación federada en dos
 fases 139
 mejorar el rendimiento 140
 Consulte también - rendimiento 230
 especificaciones de índice 271
 estadísticas de catálogo 271
 opciones de columna de apodo 240
 opciones de servidor 233
 proceso de consultas 230
 secuencias de clasificación 233
 tablas de consulta materializadas 240
almacenamiento en memoria caché 283,
 284
 apodos 315
análisis de envío
 características de apodo,
 afectación 240
 características de consulta,
 afectación 241
 características de servidor,
 afectación 233
 descripción 12, 232
 predicados con plantillas de
 función 246
aplicaciones
 apodos en 205
apodo
 métodos de recogida 177
 nombre de índice 164
 nombres de columna 164
apodos
 acceder a orígenes de datos 205
 acceso a datos 158, 162

apodos (*continuación*)
 actualizar estadísticas 182
 almacenamiento en memoria
 caché 315
 apodo para apodo 167
 crear 163
 datos XML 187
 descripción 15
 desencadenantes 158
 estadísticas 177, 180
 para un apodo 179
 recogida automática 185
 varios apodos 178
estado de actualizaciones
 Centro de control de DB2 182
EXPORT, mandato, restricciones 156
EXPORT, mandato, utilizar 155
IMPORT, mandato, ejemplos 156
IMPORT, mandato, restricciones 155
IMPORT, mandato, utilizar 155
lenguaje XQuery 187
modificación
 tipo de datos local, ejemplo 56
objetos de origen de datos válidos 16
objetos locales y remotos 157
restricciones 144
restricciones informativas 170, 171,
 172
restricciones informativas para
 apodos 171
Seguridad de etiqueta de Oracle 175
sentencias de SQL 158
tablas de memoria caché 287
WITH HOLD, sintaxis 157
archivos con estructura de tabla
 apodos, objetos válidos para 16
 características federadas
 soportadas 347
 opciones de apodo 405
 opciones de columna 405
 opciones de derivador 405
 opciones de servidor 405
 tipos de datos admitidos 458
 versiones que reciben soporte 2
archivos Excel
 apodos, objetos válidos para 16
 tipos de datos admitidos 458
 versiones que reciben soporte 2
archivos planos
 véase archivos con estructura de
 tabla 2
arquitectura
 confirmación federada en dos
 fases 118
asignación de dos fases
 orígenes de datos soportados 347
asignaciones
 federadas 148
asistente Tabla de memoria caché 288
 asistentes 288
atomicidad 124

atomicidad (*continuación*)
 preservar en sentencias 145
autenticación proxy de Oracle 345
avisos legales 467

B

base de datos DB2
 características federadas
 soportadas 347
Base de datos DB2
 opciones de apodo 353
 opciones de columna 353
 opciones de correlación de
 usuarios 353
 opciones de derivador 353
 opciones de servidor 353
base de datos federada
 derivadores 7
 módulos de derivador 7
 objetos locales 157
bases de datos federadas
 catálogo del sistema 11
 descripción 6
BioRS
 características federadas
 soportadas 347
 opciones de apodo 349
 opciones de columna 349
 opciones de correlación de
 usuarios 349
 opciones de derivador 349
 opciones de servidor 349
 tipos de datos admitidos 458

C

Capa de sockets seguros (SSL)
 orígenes de datos soportados 347
catálogo
 Consulte catálogo global 423
catálogo de herramientas de DB2 181
catálogo global 48
 actualizar estadísticas 246
 descripción 11
 vistas que contienen información
 federada 423
catálogo local
 Véase catálogo global 11
Centro de control
 interfaz para sistemas federados 8
Centro de control de DB2 9
Centro de mandatos
 utilización para federados 8
Centro de salud
 indicadores de salud 197
 instantánea de estado 199
cifrado
 descripción 295
clasificación 19

- cláusula de petición de bloqueo 214
- CLP (procesador de línea de mandatos)
 - funciones federadas 8
- colocación en antememoria 207
- compatible con VARCHAR2 234
- compensación, descripción 12
- compilador de SQL
 - diagrama de análisis de consultas 230
 - en un sistema federado 11
- conexiones fiables federadas
 - API 300, 304
 - aplicación de muestra 304
 - casos de ejemplo 301, 304, 308
 - conexiones de extremo a extremo 299, 301, 304
 - conexiones de salida fiables 299
 - conexiones fiables de salida 308
 - conexiones fiables federadas
 - conexiones entrantes 301
 - conexiones salientes 301
 - usuarios de proxy 308
 - configuración 301, 304, 308
 - correlaciones de usuarios 308, 311
 - creación
 - contextos fiables 301, 304, 308
 - definiciones de servidor 308
 - descrito 297, 299
 - diseño de aplicación 300
 - explícito 297
 - implícito
 - conexiones fiables federadas 297
 - requisitos de contraseña 301, 304, 308
 - requisitos de correlación de usuario 301, 304
 - requisitos de ID de usuario 301, 304, 308
 - tipos 299
 - usuarios de proxy federados
 - descrito 308
 - ventajas 298
- configurar
 - confirmación federada en dos fases 128
- confirmación en dos fases
 - operaciones 113
- confirmación en dos fases para transacciones federadas
 - véase confirmación federada en dos fases 117
- confirmación en una fase, operaciones de definición 113
- confirmación federada en dos fases
 - arquitectura 118
 - atomicidad 124
 - configuraciones 124
 - configurar 128
 - DDL 124
 - DDL transparente 124
 - confirmación federada en dos fases 124
 - ejemplos 120
 - habilitar 128
 - mejorar el rendimiento 140
 - operaciones permitidas 124
 - planificación 117
- confirmación federada en dos fases
 - (*continuación*)
 - rendimiento 139
 - requisitos del origen de datos 130
 - resolución de problemas 135
 - visión general 117
- conjuntos de caracteres
 - descripción 19
- conmutador de supervisor de indicación de fecha y hora 281
- conmutadores de supervisor
 - federado 281
- conmutadores del supervisor del sistema federado 281
- consultas
 - direccionar tablas de memoria caché 291
 - fragmentos 12
 - proceso asíncrono 261
 - tablas de memoria caché, direccionar 291
- contextos fiables
 - descrito 297
- contextos fiables federados
 - autenticación proxy de Oracle 345
 - orígenes de datos soportados 347
- control de acceso basado en etiquetas Federation 285
 - orígenes de datos soportados 347
 - tablas de consultas materializadas (MQT) 285
- control de acceso basado en etiquetas (LBAC)
 - federado 315
- conversión 54
- correlación de tipos de datos inversa
 - Unicode
 - orígenes de datos NET8 456
- correlaciones de funciones
 - análisis de envío, afectación 233
 - correlacionar con funciones definidas por el usuario (UDF) 63
 - correlaciones predeterminadas 61
 - crear 64
 - descripción 18, 61
 - opciones 427
- correlaciones de tipos de datos
 - análisis de envío, afectación 233
 - conversión 54
 - cuándo se deben crear 47
 - descripción 18
 - directas 431
 - descripción 49
 - en un sistema federado 48
 - inversas 444
 - descripción 49
 - no relacionales 49
 - para un objeto de origen de datos específico 56
 - para un servidor específico 53
 - para un tipo de origen de datos específico 51
 - para un tipo y versión de servidor específicos 52
 - sintaxis 49
 - situaciones que requieren
 - correlaciones nuevas 48
- correlaciones de tipos de datos
 - (*continuación*)
 - tipos de datos no soportados 47
- correlaciones de tipos de datos directas
 - correlaciones por omisión 431
 - descripción 49
 - Unicode
 - Microsoft SQL Server 454
 - orígenes de datos JDBC 453
 - orígenes de datos NET8 455
 - orígenes de datos ODBC 456
 - orígenes de datos Sybase 457
- correlaciones de tipos de datos directas por omisión
 - ejemplo de Informix 443
 - ejemplo de Microsoft SQL Server 443
 - ejemplo de Oracle 443
 - ejemplo de Sybase 444
 - ejemplo de Teradata 444
 - ejemplos 442
 - orígenes de datos DB2 Database para Linux, UNIX y Windows 431
 - orígenes de datos DB2 para System i 432
 - orígenes de datos DB2 para VM y VSE 433
 - orígenes de datos DB2 para z/OS 434
 - orígenes de datos Informix 434, 448
 - orígenes de datos JDBC 435
 - orígenes de datos Microsoft SQL Server 437, 450
 - orígenes de datos ODBC 438
 - orígenes de datos Oracle NET8 439
 - orígenes de datos Sybase 440
 - orígenes de datos Teradata 442, 453
- correlaciones de tipos de datos inversas
 - correlaciones por omisión 444
 - descripción 49
 - Unicode
 - orígenes de datos JDBC 454
 - orígenes de datos Microsoft SQL Server 455
 - orígenes de datos ODBC 456
 - orígenes de datos Sybase 457
- correlaciones de tipos de datos inversas por omisión
 - orígenes de datos DB2 Database para Linux, UNIX y Windows 445
 - orígenes de datos DB2 para System i 446
 - orígenes de datos DB2 para VM y VSE 447
 - orígenes de datos DB2 para z/OS 448
 - orígenes de datos JDBC 449
 - orígenes de datos ODBC 450
 - orígenes de datos Oracle NET8 451
 - orígenes de datos Sybase 452
- correlaciones de usuarios
 - almacenar 15, 317
 - descripción 15
 - plugins de muestra 317
 - repositorios externos 317
- CREATE FUNCTION (fuente o plantilla), sentencia 61, 67

CREATE FUNCTION MAPPING,
sentencia 61, 64
CREATE INDEX, sentencia 71
CREATE PROCEDURE (fuente), sentencia
procedimientos federados 90
CURRENT FEDERATED ASYNCHRONY,
registro especial 267

D

DB2
catálogo de herramientas 181
DB2, orígenes de datos
procedimientos federados 81
DB2 para Linux, UNIX y Windows
apodos, objetos válidos para 16
correlaciones de tipos de datos
directas por omisión 431
correlaciones de tipos de datos
inversas por omisión 444
soporte de LOB federados 217
versiones que reciben soporte 2
DB2 para System i
apodos, objetos válidos para 16
correlaciones de tipos de datos
directas por omisión 431
correlaciones de tipos de datos
inversas por omisión 444
soporte de LOB federados 217
versiones que reciben soporte 2
DB2 para VM y VSE
apodos, objetos válidos para 16
correlaciones de tipos de datos
directas por omisión 431
correlaciones de tipos de datos
inversas por omisión 444
soporte de LOB federados 217
versiones que reciben soporte 2
DB2 para z/OS
apodos, objetos válidos para 16
soporte de LOB federados 217
versiones que reciben soporte 2
DB2 para z/OS y OS/390
correlaciones de tipos de datos
directas por omisión 431
correlaciones de tipos de datos
inversas por omisión 444
DB2FEDGENTE, mandato
ejemplos 98
sintaxis 98
DDL
confirmación federada en dos
fases 124
DDL transparente
longitudes de columnas LOB 106
transacciones permitidas para 114
definiciones de servidor
descripción 14
derivador de XML 229
derivadores
descripción 7
descrito 345
Desencadenantes 158
dialecto SQL 165
análisis de envío, afectación 233
descripción 12

Distributed Relational Database
Architecture (DRDA)
configurar para confirmación federada
en dos fases 130
documentación
accesible 461, 465
DUOW
véase unidades de trabajo
distribuidas 137

E

ejemplos 150
confirmación federada en dos
fases 120
IMPORT, mandato 156
especificaciones de índice
descripción 19
federado 71
estadísticas
apodo 175, 177, 180
HIGH2KEY 181
LOW2KEY 181
métodos de recogida 177
para un apodo 179
recogida automática 185
varios apodos 178
estadísticas de HIGH2KEY 181
estadísticas de LOW2KEY 181
estadísticas federadas
actualizar 175
estado de actualizaciones
apodos 182
Excel
características federadas
soportadas 347
opciones de apodo 360
opciones de derivador 360
opciones de servidor 360
EXPORT, mandato
apodos, utilizar con 155, 156
expresiones de tabla anidadas
tolerancia a errores 193

F

función XMLVALIDATE 189
funciones
plugin de correlación de usuario 317
funciones definidas por el usuario
(UDF) 18
en aplicaciones de sistema
federado 63
transacciones permitidas para 114
funciones incorporadas 18
funciones SQL/XML 187

G

grupos de particiones
computacionales 257

H

habilitar
confirmación federada en dos
fases 128
herramienta db2exfmt
visualización de planes de
acceso 242, 277
herramienta db2expln
visualización de planes de
acceso 242, 277
herramienta dynexpln
visualización de planes de
acceso 242, 277
herramientas, catálogo de 181
herramientas de Explain 259
heurísticas, operaciones
resolver transacciones dudosas
sistemas federados 136
horas, campo 150
horas, con 24 en el campo de horas 150

I

IMPORT, mandato
apodos, ejemplos 156
apodos, restricciones 155
apodos, utilizar con 155
importación de datos
orígenes de datos soportados 347
indicaciones de fecha y hora
con 24 en el campo de horas 150
indicadores de salud federados
orígenes de datos soportados 347
información de catálogo remota 11
Informix
apodos, objetos válidos para 16
características federadas
soportadas 347
configurar para confirmación federada
en dos fases 132
correlaciones de tipos de datos
directas por omisión 431
correlaciones de tipos de datos
inversas por omisión 444
opciones de columna 361
opciones de correlación de
usuarios 361
opciones de derivador 361
opciones de servidor 361
soporte de LOB federados 217
versiones que reciben soporte 2
InfoSphere Data Architect 10
integridad referencial 144
IUD_APP_SVPT_ENFORCE, opción de
servidor
ejemplos 145

J

JDBC
apodos, objetos válidos para 16
características federadas
soportadas 347
opciones de columna 367
opciones de correlación de
usuarios 367

JDBC (*continuación*)
opciones de derivador 367
opciones de servidor 367
soporte de LOB federados 217
versiones que reciben soporte 2

L

LBAC
Federation 285
tablas de consultas materializadas (MQT) 285
lectores de pantalla 465
lectores de pantallas 461
LOB
orígenes de datos soportados, lectura y escritura 347
orígenes de datos soportados, sólo lectura 347
LOB (objeto grande), tipos de datos actualización, operaciones de 145

M

marcas registradas 469
Microsoft Excel
véase archivos Excel 2
Microsoft SQL Server
apodos, objetos válidos para 16
características federadas soportadas 347
configurar para confirmación federada en dos fases 133
correlaciones de tipos de datos directas por omisión 431
correlaciones de tipos de datos directas por omisión Unicode 454
correlaciones de tipos de datos inversas por omisión 444
opciones de columna 374
opciones de correlación de usuarios 374
opciones de derivador 374
opciones de servidor 374
soporte de LOB federados 217
versiones que reciben soporte 2
Microsoft SQL Server, orígenes de datos procedimientos federados 82
miembro de conjunto de suscripción tablas de memoria caché 291
modificación 24
tipos de datos LONG 58
MQT (tablas de consultas materializadas) en apodos 240
federadas
visión general 283
restricciones de apodos 285
seguridad de etiquetas de Oracle (OLS) 285

N

niveles de aislamiento
sistemas federados 213

O

objetos de origen de datos 162
descripción 15
tipos de objeto válidos 16
objetos grandes (LOB), tipos de datos actualización, operaciones de 145
objetos locales 157
objetos remotos, apodos 157
ODBC
apodos, objetos válidos para 16
características federadas soportadas 347
correlaciones de tipos de datos directas por omisión 431
opciones de columna 379
opciones de correlación de usuarios 379
opciones de derivador 379
opciones de servidor 379
soporte de LOB federados 217
versiones que reciben soporte 2
OLE DB
versiones que reciben soporte 2
OLS (seguridad de etiqueta de Oracle) restricciones de apodos tablas de consultas materializadas (MQT) 285
opción DB2_UM_PLUGIN 328
opción DB2_UM_PLUGIN_LANG 328
opción de columna NUMERIC_STRING oportunidades de envío, afectación 240
opción de columna VARCHAR_NO_TRAILING_BLANKS oportunidades de envío, afectación 240
opción de correlaciones de funciones DISABLE valores válidos 427
opción de correlaciones de funciones REMOTE_NAME valores válidos 427
opción de servidor COLLATING_SEQUENCE ejemplo 19
oportunidades de envío, afectación 233
optimización global, afectación 271
opción de vinculación FEDERATED_ASYNCHRONY 267
opción del servidor COMM_RATE optimización global, afectación 271
opción del servidor CPU_RATIO optimización global, afectación 271
opción del servidor DB2_MAX_ASYNC_REQUESTS_PER_QUERY 267
opción del servidor DB2_MAXIMAL_PUSHDOWN decisiones de análisis de envío 242
oportunidades de envío, afectación 233
opción del servidor IO_RATIO optimización global, afectación 271
opción del servidor PLAN_HINTS optimización global, afectación 271

opción del servidor VARCHAR_NO_TRAILING_BLANKS oportunidades de envío, afectación 233
opciones de apodo
archivos con estructura de tabla 405
Base de datos DB2 353
BioRS 349
Script 391
servicios web 408
XML 416
opciones de columna
análisis de envío, afectación 240
archivos con estructura de tabla 405
Base de datos DB2 353
BioRS 349
descripción 17
Informix 361
JDBC 367
Microsoft SQL Server 374
ODBC 379
Oracle 385
Script 391
servicios web 408
Sybase 396
Teradata 401
XML 416
opciones de columna de apodo descripción 17
opciones de correlación de usuarios
Base de datos DB2 353
BioRS 349
Informix 361
JDBC 367
Microsoft SQL Server 374
ODBC 379
Oracle 385
Script 391
servicios web 408
Sybase 396
Teradata 401
XML 416
opciones de derivador
archivos con estructura de tabla 405
Base de datos DB2 353
BioRS 349
Excel 360
Informix 361
JDBC 367
Microsoft SQL Server 374
ODBC 379
Oracle 385
Script 391
servicios web 408
Sybase 396
Teradata 401
XML 416
opciones de servidor
análisis de envío, afectación 233
archivos con estructura de tabla 405
asincronía 267
Base de datos DB2 353
BioRS 349
descripción 14
Excel 360
Informix 361
JDBC 367

- opciones de servidor *(continuación)*
 - Microsoft SQL Server 374
 - ODBC 379
 - optimización global, afectación 271
 - Oracle 385
 - Script 391
 - servicios web 408
 - Sybase 396
 - temporal 14
 - Teradata 401
 - XML 416
- operaciones de escritura
 - Vea actualizaciones 114
- operador GROUP BY
 - decisiones de evaluación de plan de acceso 243
 - decisiones de optimización de planes de acceso 278
- operador ORDER BY
 - decisiones de evaluación de plan de acceso 243
- operadores set
 - decisiones de evaluación de plan de acceso 243
- Optim Data Studio 10
- optimización
 - características de servidor, afectación 271
 - consultas asíncronas 268
- optimización asíncrona
 - orígenes de datos soportados 347
- optimización de consultas
 - descripción 12
- optimización global 246
 - características de servidor, afectación 271
 - descripción 271
- optimizador
 - descripción 12
- Oracle 345
 - apodos, objetos válidos para 16
 - características federadas soportadas 347
 - configurar para confirmación federada en dos fases 131
 - correlaciones de tipos de datos directas por omisión 431
 - correlaciones de tipos de datos inversas por omisión 444
 - opciones de columna 385
 - opciones de correlación de usuarios 385
 - opciones de derivador 385
 - opciones de servidor 385
 - resolución de problemas de la confirmación federada en dos fases 138
 - series vacías 150
 - soporte de LOB federados 217
- Oracle, orígenes de datos
 - procedimientos con sobrecarga 86
 - procedimientos federados 84
- origen de datos
 - semántica de VARCHAR2 234
- origen de datos Oracle
 - seguridad 345
- orígenes de datos 6, 11
 - orígenes de datos *(continuación)*
 - características soportadas 347
 - crear apodos 163
 - descripción 6
 - indicaciones de plan remoto y rendimiento 271
 - opciones 349
 - requisitos para la confirmación federada en dos fases 130
 - secuencia de clasificación y rendimiento 271
 - tipos de servidor válidos 429
 - velocidad de comunicación y rendimiento 271
 - velocidad de E/S y rendimiento 271
 - velocidad de procesador y rendimiento 271
 - orígenes de datos DB2 Database para Linux, UNIX y Windows
 - correlaciones de tipos de datos directas por omisión 431
 - correlaciones de tipos de datos inversas por omisión 445
 - orígenes de datos DB2 para System i
 - correlaciones de tipos de datos directas por omisión 432
 - correlaciones de tipos de datos inversas por omisión 446
 - orígenes de datos DB2 para VM y VSE
 - correlaciones de tipos de datos directas por omisión 433
 - correlaciones de tipos de datos inversas por omisión 447
 - orígenes de datos DB2 para z/OS
 - correlaciones de tipos de datos directas por omisión 434
 - correlaciones de tipos de datos inversas por omisión 448
 - orígenes de datos Informix
 - correlaciones de tipos de datos directas por omisión 434, 448
 - orígenes de datos JDBC
 - correlación de tipos de datos directa por omisión Unicode 453
 - correlación de tipos de datos inversa por omisión Unicode 454
 - correlaciones de tipos de datos directas por omisión 435
 - correlaciones de tipos de datos inversas por omisión 449
 - orígenes de datos Microsoft SQL Server
 - correlación de tipos de datos inversa por omisión Unicode 455
 - correlaciones de tipos de datos directas por omisión 437, 450
 - orígenes de datos NET8
 - correlación de tipos de datos inversa por omisión Unicode 456
 - correlaciones de tipos de datos directas por omisión Unicode 455
 - orígenes de datos no relacionales admitidos, tipos de datos 458
 - especificación de correlaciones de tipos de datos 18
 - orígenes de datos ODBC
 - correlación de tipos de datos inversa por omisión Unicode 456
 - orígenes de datos ODBC *(continuación)*
 - correlaciones de tipos de datos directas por omisión 438
 - correlaciones de tipos de datos directas por omisión Unicode 456
 - correlaciones de tipos de datos inversas por omisión 450
 - orígenes de datos Oracle NET8
 - correlaciones de tipos de datos directas por omisión 439
 - correlaciones de tipos de datos inversas por omisión 451
 - orígenes de datos Sybase
 - correlación de tipos de datos inversa por omisión Unicode 457
 - correlaciones de tipos de datos directas por omisión 440
 - correlaciones de tipos de datos directas por omisión Unicode 457
 - correlaciones de tipos de datos inversas por omisión 452
 - orígenes de datos Teradata
 - correlaciones de tipos de datos directas por omisión 442, 453

P

- páginas de códigos
 - descripción 19
- paralelismo 249, 257, 258, 259
 - federado 249, 251
- paralelismo dentro de particiones 249
 - federadas 249
 - planes de acceso federados 250
- paralelismo entre particiones 249
 - federadas 251, 254, 257, 258
- paralelismo mixto
 - orígenes de datos federados
 - plan de acceso 259
 - proceso de datos 259
 - visión general 249
- parámetro de configuración FEDERATED_ASYNC 267
- paso a través
 - descripción 13
 - restricciones 13
 - Soporte de LOB 218
 - transacciones permitidas para 114
- plan de acceso 259
- planes de acceso 254
 - decisiones de evaluación 243
 - decisiones de optimización 278
 - descripción 12
 - optimización de asincronía 263
 - rendimiento 278
 - visualización 242, 277
- planificación
 - confirmación federada en dos fases 117
- plantillas de función
 - descripción 67
 - envío de predicado 246
- plugins de correlación de usuario
 - acceso de servidor federado 328
 - lenguaje C
 - creación 327
 - denominación 328

- plugins de correlación de usuario (*continuación*)
 - lenguaje de programación C
 - actualizar 329
 - archivo de cabecera 320
 - declaración de funciones 325
 - declaraciones de función 324
 - desarrollo 324
 - despliegue 328
 - función FSUMconnect 323
 - función FSUMdisconnect 324
 - función FSUMfetchUM 323
 - función FSUMPluginInit 322
 - función FSUMPluginTerm 324
 - funciones 317
 - manejo de errores 326
 - plataformas soportadas 319
 - plugin de muestra 329
 - prueba 327
 - punteros de función 325
 - restricciones 319
 - visión general 317
 - lenguaje de programación Java
 - archivo de configuración 341
 - archivos de muestra 336, 339, 340
 - arquitectura 330
 - clases 332, 333, 334, 335
 - compilación de archivos 341
 - configuración de acceso 343
 - desarrollo 337, 338
 - despliegue 343
 - prueba 342
 - opción DB2_UM_PLUGIN 328
 - opción
 - DB2_UM_PLUGIN_LANG 328
- predicados
 - con plantillas de función 246
 - decisiones de evaluación de plan de acceso 243
- procedimiento almacenado
 - SYSPROC.NNSTAT 183
- procedimientos
 - federados 79
 - federados, crear 90
 - federados, eliminar 94
 - federados, resolución de problemas 99
 - federados, unir conjuntos de resultados 95
- procedimientos almacenados
 - estadísticas de apodo 183
- procedimientos con sobrecarga
 - procedimientos federados 86
- procedimientos federados 19
 - crear 90
 - CREATE PROCEDURE (fuente), sentencia 90
 - DB2 81
 - eliminar 94
 - llamar 93
 - Microsoft SQL Server 82
 - Oracle 84
 - orígenes de datos soportados 347
 - parámetros 93
 - privilegios, otorgar 91
 - privilegios, revocar 91
 - resolución de problemas 99

- procedimientos federados (*continuación*)
 - soporte de origen de datos 81
 - Sybase 88
 - unir conjuntos de resultados 95
 - visión general 79
 - vistas de catálogo 93
- procesador de línea de mandatos (CLP)
 - funciones federadas 8
- proceso asíncrono
 - descripción 261
 - ejemplos 261
 - habilitar 267
 - optimización 269
 - restricciones 268
- programas de aplicación 63
- Proxy HTTP
 - orígenes de datos soportados 347
- proxy SOCKS
 - orígenes de datos soportados 347
- puntos de rescate
 - API de origen de datos 145
 - en federación 152
- puntos de rescate de aplicación 152
 - orígenes de datos soportados 347

R

- recurso de actualización de estadísticas de apodo
 - orígenes de datos soportados 347
- Recurso Explain 250
- Registro especial CURRENT IMPLICIT XMLPARSE OPTION 191
- reglas
 - semántica de asignación en sistema federado 148
- rendimiento 170, 232, 241, 257, 259
 - confirmación federada en dos fases 139
 - mejorar 140
- Consulte también - ajuste 230
- diferencias de SQL 233
- federadas 175, 180
- federado 172, 229
- federados 183
- indicaciones de plan remoto 271
- proceso de consultas asíncrono 261
- secuencia de clasificación 271
- secuencias de clasificación 233
- velocidad de comunicación 271
- velocidad de CPU 271
- velocidad de E/S 271
- repositorio de correlación de usuarios
 - orígenes de datos soportados 347
- requisitos 130
- requisitos del origen de datos para la confirmación federada en dos fases 130
- resolución de problemas
 - confirmación federada en dos fases 135, 138
- resolución de problemas de la confirmación federada en dos fases 138
- restricciones informativas
 - apodos 170, 172

- restricciones informativas para apodos 172
- RETURN DATA UNTIL, cláusula 194

S

Script

- características federadas soportadas 347
- opciones de apodo 391
- opciones de columna 391
- opciones de correlación de usuarios 391
- opciones de derivador 391
- opciones de servidor 391
- versiones que reciben soporte 2

secuencias de clasificación

- descripción 19
- planificación 19
- visión general 233

seguridad 345

Seguridad

- Oracle 345

Seguridad de etiqueta de Oracle

- descripción 345

seguridad de etiquetas de Oracle (OLS)

- restricciones de apodos
 - tablas de consultas materializadas (MQT) 285

semántica de asignación federada

- ejemplos 150

sentencia ALTER NICKNAME

- ejemplo
 - tipo de datos local 56

sentencia ALTER WRAPPER 24

sentencia CREATE INDEX 19

sentencia CREATE NICKNAME 49

sentencia CREATE SERVER 5

sentencia CREATE TYPE MAPPING 48, 51, 52, 53

sentencia DELETE 144

- decisiones de evaluación de plan de acceso 243

sentencia INSERT 143, 144

- decisiones de evaluación de plan de acceso 243

sentencia SET SERVER OPTION

- establecimiento de una opción temporalmente 14

sentencia UPDATE 144

- decisiones de evaluación de plan de acceso 243

sentencias de SQL

- apodos 158

series

- secuencias de clasificación 19
- vacío, utilizado con Oracle 150

servicios web

- características federadas soportadas 347
- opciones de apodo 408
- opciones de columna 408
- opciones de correlación de usuarios 408
- opciones de derivador 408
- opciones de servidor 408
- tipos de datos admitidos 458

servidor federado 6
 descripción 5
 semántica de VARCHAR2 234
 servidor LDAP
 correlaciones de usuarios 317
 servidores proxy
 descrito 295
 sesiones de paso a través 165
 orígenes de datos soportados 347
 sistema de gestión de bases de datos
 distribuidas 1
 sistemas federados
 aislamiento de nivel de conexión 215
 aislamiento de nivel de sentencia 214
 niveles de aislamiento 213
 visión general 1
 SQL Explain
 visualización de planes de
 acceso 242, 277
 supervisar la instantánea 200
 apodos y servidores 199
 fragmentos de consulta
 federados 202
 servidores y apodos federados 197
 Sybase 138
 apodos, objetos válidos para 16
 características federadas
 soportadas 347
 configurar para confirmación federada
 en dos fases 133
 correlaciones de tipos de datos
 directas por omisión 431
 correlaciones de tipos de datos
 inversas por omisión 444
 opciones de columna 396
 opciones de correlación de
 usuarios 396
 opciones de derivador 396
 opciones de servidor 396
 soporte de LOB federados 217
 versiones que reciben soporte 2
 Sybase, orígenes de datos
 procedimientos federados 88
 SYSPROC.FED_STATS, tabla 182

T

tablas de consulta materializadas
 adición a tablas de memoria
 caché 290
 eliminación de tablas de memoria
 caché 292
 orígenes de datos soportados 347
 tablas de memoria caché 287
 tablas de consultas materializadas
 (MQT) 207, 284
 en apodos 240
 federadas
 visión general 283
 restricciones de apodos 285
 tablas de memoria caché 285
 alterar 289
 asistente Tabla de memoria
 caché 288
 crear 288
 descripción 287
 direccionar consultas 291

tablas de memoria caché (*continuación*)
 eliminación 293
 habilitar 291
 inhabilitar 291
 modificar 289
 orígenes de datos 288
 orígenes de datos soportados 347
 planificaciones de réplicas 287
 registro de archivado 288
 réplica 287
 requisitos previos 288
 tablas de consulta materializadas 290
 ver valores 289
 Teradata
 apodos, objetos válidos para 16
 características federadas
 soportadas 347
 correlaciones de tipos de datos
 directas por omisión 431
 correlaciones de tipos de datos
 inversas por omisión 444
 opciones de columna 401
 opciones de correlación de
 usuarios 401
 opciones de derivador 401
 opciones de servidor 401
 soporte de LOB federados 217
 TIMESTAMP 55
 tipo de datos DATALINK
 no soportados 18
 tipos de datos 106
 análisis de envío, afectación 240
 no soportados 18
 para orígenes de datos no
 relacionales 458
 semántica de VARCHAR2 234
 TIMESTAMP 55
 tipos de datos de LOB (objeto grande)
 localizadores 218
 restricciones 218
 tipos de datos de objeto grande (LOB)
 consideraciones de rendimiento 219
 localizadores 218
 restricciones 218
 tipos de datos LONG 58
 tipos de servidor
 tipos federados válidos 429
 tipos definidos por el usuario (UDT)
 tipos de datos no soportados 18
 tolerancia a errores
 descripción 193
 ejemplo 195
 habilitar 194
 restricciones 196
 soporte de orígenes de datos 196
 tolerancia de errores
 orígenes de datos soportados 347
 transacciones
 actualizaciones 114
 visión general 113
 transacciones dudosas
 rastrear unidades de trabajo
 distribuidas 137
 recuperación 135
 resincronizar para sistemas
 federados 135

transacciones dudosas (*continuación*)
 resolver
 sistemas federados 136

U

Unicode
 orígenes de datos soportados 347
 unidades de trabajo distribuidas
 rastrear entre orígenes de datos 137
 uniones
 decisiones de optimización de planes
 de acceso 278
 unir conjuntos de resultados
 DB2FEDGENTF, ejemplos del
 mandato 98
 DB2FEDGENTF, sintaxis del
 mandato 98

V

VARCHAR2
 semántica 234
 visión general
 confirmación federada en dos
 fases 117
 vistas de catálogo SYSCAT 63, 423
 vistas de catálogo SYSSTAT 423
 vistas federadas
 acceso a datos 165
 crear 166
 Visual Explain
 visualización de planes de
 acceso 242, 277

W

WITH HOLD, opción 157
 WITH HOLD, sintaxis 157

X

XML
 apodos, objetos válidos para 16
 documentos
 descomposición 190
 validación 189
 esquemas, registro de 188
 opciones de apodo 416
 opciones de columna 416
 opciones de correlación de
 usuarios 416
 opciones de derivador 416
 opciones de servidor 416
 repositorio de esquemas 190
 tipo de datos
 restricciones 191
 soporte 187
 tipos de datos admitidos 458
 versiones que reciben soporte 2



Impreso en España

SC11-3953-02



Spine information:

IBM InfoSphere Federation Server

Versión 9.7

Guía de administración para sistemas federados

