

IBM DB2 Information Integrator



データ・ソース構成ガイド

バージョン 8.2

IBM DB2 Information Integrator



データ・ソース構成ガイド

バージョン 8.2

ご注意！

本書および本書で紹介する製品をご使用になる前に、『特記事項』に記載されている情報をお読みください。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： IBM DB2 Information Integrator
Data Source Configuration Guide
Version 8.2

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2004.8

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 1998, 2004. All rights reserved.

© Copyright IBM Japan 2004

目次

本書について	xi
本書の対象読者	xi
本書の規則および用語	xi
構文図の読み方	xii

第 1 部 概念および計画 1

第 1 章 フェデレーテッド・システムの概要 3

フェデレーテッド・システム	3
フェデレーテッド・サーバー	4
データ・ソースとは?	5
サポートされるデータ・ソース	6
フェデレーテッド・データベース	8
フェデレーテッド・データベース・システム・カタログ	9
SQL コンパイラー	10
照会オプティマイザー	10
適合	11
パススルー・セッション	12
ラッパーおよびラッパー・モジュール	13
デフォルトのラッパー名	14
サーバー定義およびサーバー・オプション	15
ユーザー・マッピング	16
ニックネームとデータ・ソース・オブジェクト	17
有効なデータ・ソース・オブジェクト	18
ニックネーム列オプション	19
データ・タイプ・マッピング	20
関数マッピング	21
索引の指定	21
フェデレーテッド・システムと対話する方法	22
DB2 コマンド行プロセッサ (CLP)	22
DB2 コマンド・センター	23
DB2 コントロール・センター	23
アプリケーション・プログラム	24
DB2 ファミリー・ツール	24
Web サービス・プロバイダー	24

第 2 章 フェデレーテッド・データ・ソースの構成の計画 25

フェデレーテッド・オブジェクトの命名規則	25
フェデレーテッド・システムでの大文字小文字の区別をする値の保持	26
データ・ソース統計の更新	27
正しいラッパーの選択	28
Excel データにアクセスする方法	29
ユーザー・マッピングの計画	33
データ・タイプ・マッピングの計画	33
関数マッピングの計画	35

フェデレーテッド・システム構成の計画のためのチェックリスト	35
チェックリスト: フェデレーテッド・オブジェクトのネーミング規則	35
チェックリスト: 大文字小文字の区別をする値の保持	35
チェックリスト: データ・ソース統計	36
チェックリスト: データ・タイプ・マッピング	36
チェックリスト: ユーザー・マッピング	37
チェックリスト: ラッパー	37

第 2 部 フェデレーテッド・サーバーおよびデータベース 41

第 3 章 フェデレーテッド・サーバーのセットアップのチェック 43

フェデレーテッド・サーバーのセットアップのチェック	43
ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)	43
ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)	43
ラッパー・ライブラリー・ファイルのチェック (UNIX)	44
ラッパー・ライブラリー・ファイル	45
リンク・エディット・メッセージ・ファイルのチェック (UNIX)	46
データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする	47
FEDERATED パラメーターのチェック	49

第 4 章 フェデレーテッド・データベースの作成 51

フェデレーテッド・データベースの各国語に関する考慮事項	51
フェデレーテッド・システムの照合シーケンス	52
フェデレーテッド・データベース照合シーケンスの設定	54
フェデレーテッド・システムの Unicode サポート	55
フェデレーテッド・データベースの作成	57

第 3 部 データ・ソース 59

第 5 章 データ・ソースへのアクセスの構成の概要 61

データ・ソースの構成の近道	61
DB2 UDB コントロール・センターを使用したフェデレーテッド・サーバーへのデータ・ソースの追加	63

データ・ソースへアクセスするための複数のフェデ	
レーテッド・サーバーの構成	64
データ・ソース環境変数の設定	65
データ・ソース環境変数の設定	65
db2dj.ini ファイルの制約事項	66
複数パーティション・インスタンス構成での環境	
変数の適用	67
データ・ソースのラッパーの登録	68
データ・ソースのサーバー定義の登録	69
データ・ソースのユーザー・マッピングの登録	70
データ・ソースのニックネームの登録	71
データ・ソースのニックネームの登録	71
ニックネームに対するニックネームの作成	73
非リレーショナル・データ・ソースのニックネー	
ム列の指定	74
オプションの構成ステップ	74

第 6 章 BioRS データ・ソースへのア

クセスの構成	77
BioRS とは何か	77
フェデレーテッド・サーバーへの BioRS の追加	78
フェデレーテッド・サーバーへの BioRS デー	
タ・ソースの追加	78
BioRS ラッパーのカスタム関数の登録	79
カスタム関数表 - BioRS ラッパー	80
BioRS ラッパーの登録	81
BioRS ラッパーのライブラリー・ファイル	82
BioRS データ・ソースのサーバー定義の登録	82
BioRS データ・ソースのユーザー・マッピングの	
登録	83
BioRS データ・ソースのニックネームの登録	84
CREATE NICKNAME ステートメント - BioRS	
ラッパーの例	86
BioRS データ・ソースの照会およびカスタム関数	87
カスタム関数と BioRS 照会	87
BioRS ラッパーの等価結合述部	90
BioRS AllText エレメント	91
BioRS ラッパー - 照会例	92
BioRS ラッパー・パフォーマンスの最適化	97
BioRS ラッパー・パフォーマンスの最適化方法	97
BioRS 統計情報	98
BioRS データ・バンク・カーディナリティー統計	
の確認	99
BioRS ニックネーム・カーディナリティー統計の	
更新	99
BioRS 列カーディナリティー統計	100
BioRS _ID_ 列カーディナリティーの更新	101
BioRS ラッパーのメッセージ	102

第 7 章 BLAST データ・ソースへのア

クセスの構成	107
BLAST とは?	107
フェデレーテッド・サーバーへの BLAST の追加	111
フェデレーテッド・サーバーへの BLAST デー	
タ・ソースの追加	111

blastall 実行可能ファイルとマトリックス・ファ	
イルの正しいバージョンがインストールされてい	
ることの検証	112
BLAST デーモンの構成	113
BLAST デーモンの構成ファイル - 例	116
BLAST デーモンの開始	117
db2blast_daemon コマンド - 構文および例	118
BLAST ラッパーの登録	119
BLAST ラッパーのライブラリー・ファイル	120
BLAST データ・ソースのサーバーの登録	121
BLAST データ・ソースのニックネームの登録	121
BLAST ラッパーで作業するための TurboBlast のセ	
ットアップ	128
BLAST SQL 照会の作成	129
BLAST データ・ソース - 照会の例	130
BLAST ラッパーの最適化のヒント	132
BLAST ラッパーのメッセージ	132

第 8 章 ビジネス・アプリケーション・

データ・ソースへのアクセスの構成	135
WebSphere Business Integration ラッパー	135
ビジネス・オブジェクト定義	137
WebSphere Business Integration アダプターの構成	139
フェデレーテッド・サーバーへのビジネス・アプリ	
ケーションの追加	142
ビジネス・アプリケーション・データ・ソースの	
フェデレーテッド・システムへの追加	142
WebSphere Business Integration ラッパーの登録	143
WebSphere Business Integration ラッパー・ライ	
ブラリー・ファイル	144
ビジネス・アプリケーション・データ・ソースの	
サーバー定義の登録	145
ビジネス・アプリケーション・データ・ソースの	
ニックネームの登録	146
ビジネス・アプリケーション・ラッパーおよび	
Web サービス・ラッパーの照会制約事項	169
ビジネス・アプリケーション・データ・ソース - 照	
会の例	173

第 9 章 DB2 ファミリー・データ・ソ

ースへのアクセスの構成	175
フェデレーテッド・サーバーへの DB2 ファミリー	
ー・データ・ソースの追加	175
フェデレーテッド・サーバーへの DB2 ファミリー	
ー・データ・ソースの追加	175
フェデレーテッド・ノード・ディレクトリーへの	
ノード項目のカタログ	176
フェデレーテッド・システム・データベース・デ	
ィレクトリーへのリモート・データベースのカタ	
ログ	177
DB2 ラッパーの登録	178
DB2 ラッパーのライブラリー・ファイル	179
DB2 データ・ソースのサーバー定義の登録	180
CREATE SERVER ステートメント - DB2 ラッ	
パーの例	181

DB2 データ・ソースのユーザー・マッピングの作成	182
CREATE USER MAPPING ステートメント - DB2 ラッパーの例	183
DB2 データ・ソース・サーバーへの接続のテスト	184
DB2 表およびビューのニックネームの登録	185
CREATE NICKNAME ステートメント - DB2 ラッパーの例	186

第 10 章 Documentum データ・ソースへのアクセスの構成 189

Documentum とは?	189
フェデレーテッド・サーバーへの Documentum の追加	191
フェデレーテッド・サーバーへの Documentum データ・ソースの追加	191
Documentum クライアント・ライブラリーのラッパーでの使用可能化	192
Documentum 環境変数の設定	193
Documentum ラッパーの登録	194
Documentum ラッパーのライブラリー・ファイル	195
Documentum データ・ソースのサーバーの登録	195
Documentum データ・ソースのユーザー・マッピングの登録	196
Documentum データ・ソースのニックネームの登録	197
Documentum データ・ソースの照会およびカスタム関数	204
Documentum ラッパーのカスタム関数の登録	204
Documentum データ・ソース - 照会の例	209
Documentum ラッパーのアクセス・コントロール	211
Documentum ラッパーのメッセージ	211

第 11 章 Entrez データ・ソースへのアクセスの構成 217

Entrez とは?	217
フェデレーテッド・サーバーへの Entrez の追加	218
フェデレーテッド・サーバーへの Entrez データ・ソースの追加	218
Entrez ラッパーのカスタム関数の登録	219
カスタム関数表 - Entrez ラッパー	220
Entrez ラッパーの登録	220
Entrez ラッパーのライブラリー・ファイル	221
Entrez データ・ソースのサーバーの登録	222
Entrez データ・ソースのニックネームの登録	224
Entrez データ・ソースの照会およびカスタム関数	226
カスタム関数と Entrez 照会	226
Entrez ラッパーの関係述部	227
Entrez ラッパーで無効な WHERE 文節	228
スキーマ・データ・エレメントの単純化	228
Entrez データ・ソース - 照会の例	229
PubMed スキーマ表	230
Nucleotide スキーマ表	235
Entrez ラッパーのメッセージ	239

第 12 章 Excel データ・ソースへのアクセスの構成 243

Excel とは?	243
フェデレーテッド・サーバーへの Excel の追加	244
フェデレーテッド・サーバーへの Excel データ・ソースの追加	244
Excel ラッパーの登録	245
Excel ラッパーのライブラリー・ファイル	246
Excel データ・ソースのサーバーの登録	247
Excel データ・ソースのニックネームの登録	247
Excel データ・ソース - 照会の例	248
Excel データ・ソース - サンプル・シナリオ	249
Excel ラッパーのファイル・アクセス・コントロール・モデル	250
Excel ラッパーのメッセージ	251

第 13 章 Extended Search データ・ソースへのアクセスの構成 257

Extended Search とは?	257
Extended Search のデータ・ソース	258
Extended Search ラッパーがどのように機能するか	258
Extended Search のニックネーム	260
Extended Search の垂直表	261
フェデレーテッド・サーバーへの Extended Search の追加	264
フェデレーテッド・サーバーへの Extended Search データ・ソースの追加	264
Extended Search ラッパーの登録	265
Extended Search ラッパーのライブラリー・ファイル	266
Extended Search データ・ソースのサーバーの登録	266
Extended Search データ・ソースのユーザー・マッピングの登録	267
Extended Search データ・ソースのニックネームの登録	268
Extended Search ラッパーのカスタム関数の登録	269
Extended Search データ・ソースの照会	270
Extended Search ラッパー - 照会のガイドライン	270
Extended Search ラッパー - 照会の例	272
Extended Search ラッパー - 汎用照会言語	274
Extended Search ラッパーのメッセージ	276

第 14 章 HMMER データ・ソースへのアクセスの構成 281

HMMER とは?	281
フェデレーテッド・サーバーへの HMMER の追加	284
フェデレーテッド・サーバーへの HMMER データ・ソースの追加	284
HMMER プログラムの実行可能ファイルのパージョンの検査	285
HMMER デーモンの構成	286
HMMER デーモン構成ファイル - 例	290
HMMER デーモンの開始	290

db2hmmmer_daemon コマンド - 構文および例	292
HMMER ラッパーの登録	293
HMMER ラッパー・ライブラリー・ファイル	294
HMMER データ・ソースのサーバー定義の登録	294
CREATE SERVER ステートメント - HMMER	
ラッパーの例	295
HMMER データ・ソースのニックネームの登録	296
HMMER ニックネームの固定列	297
CREATE NICKNAME ステートメント -	
HMMER ラッパーの例	301
HMMER データ・ソース - 完全指定の例	302
サンプルを使用した新規 HMMER 照会の構成	304
HMMER ラッパーのメッセージ	305

第 15 章 Informix データ・ソースへのアクセスの構成 307

フェデレーテッド・サーバーへの Informix の追加	307
フェデレーテッド・サーバーへの Informix データ・ソースの追加	307
Informix クライアント構成ファイルのセットアップおよびテスト	308
Informix 環境変数の設定	309
Informix ラッパーの登録	313
Informix ラッパー・ライブラリー・ファイル	314
Informix データ・ソースのサーバー定義の登録	315
CREATE SERVER ステートメント - Informix	
ラッパーの例	316
Informix データ・ソースのユーザー・マッピングの作成	318
CREATE USER MAPPING ステートメント -	
Informix ラッパーの例	318
Informix サーバーへの接続のテスト	319
Informix 表、ビュー、およびシノニムのニックネームの登録	320
CREATE NICKNAME ステートメント -	
Informix ラッパーの例	321
Informix データ・ソースに対する構成のチューニングおよびトラブルシューティング	322
FOLD_ID および FOLD_PW サーバー・オプションを設定してパフォーマンスを改善する	322

第 16 章 Microsoft SQL Server データ・ソースへのアクセスの構成 325

フェデレーテッド・サーバーへの Microsoft SQL Server の追加	325
フェデレーテッド・サーバーへの Microsoft SQL Server データ・ソースの追加	325
Microsoft SQL Server データ・ソースにアクセスするためのフェデレーテッド・サーバーの準備	326
Microsoft SQL Server 環境変数の設定	328
Microsoft SQL Server ラッパーの登録	330
Microsoft SQL Server ラッパー・ライブラリー・ファイル	331
Microsoft SQL Server データ・ソースのサーバー定義の登録	332

CREATE SERVER ステートメント - Microsoft SQL Server ラッパーの例	333
Microsoft SQL Server データ・ソースのユーザー・マッピングの作成	335
CREATE USER MAPPING ステートメント -	
Microsoft SQL Server ラッパーの例	335
Microsoft SQL Server リモート・サーバーへの接続のテスト	336
Microsoft SQL Server 表およびビューのニックネームの登録	338
CREATE NICKNAME ステートメント -	
Microsoft SQL Server ラッパーの例	339
Microsoft SQL Server データ・ソースのための構成のチューニングとトラブルシューティング	339
ODBC トレースの入手	339

第 17 章 ODBC データ・ソースへのアクセスの構成 341

ODBC のフェデレーテッド・システムへの追加	341
フェデレーテッド・サーバーへの ODBC データ・ソースの追加	341
ODBC を通じてデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備	343
ODBC ラッパーの登録	344
ODBC ラッパー・ライブラリー・ファイル	345
CREATE WRAPPER ステートメント - ODBC	
ラッパーの例	345
ODBC データ・ソースのサーバー定義の登録	346
CREATE SERVER ステートメント - ODBC	
ラッパーの例	347
ODBC データ・ソースのユーザー・マッピングの作成	348
CREATE USER MAPPING ステートメント -	
ODBC ラッパーの例	349
ODBC データ・ソース・サーバーへの接続のテスト	350
ODBC データ・ソース表およびビューのニックネームの登録	351
CREATE NICKNAME ステートメント - ODBC	
ラッパーの例	353
ODBC ラッパーを使用した Excel データへのアクセス	353
デフォルトのデータ・タイプ・マッピングの変更	355
ALTER SERVER ステートメント - ODBC	
ラッパーで Excel データにアクセスする例	356
ODBC データ・ソースのための構成のチューニングとトラブルシューティング	357
ODBC トレースの入手	357

第 18 章 OLE DB データ・ソースへのアクセスの構成 359

フェデレーテッド・サーバーへの OLE DB データ・ソースの追加	359
OLE DB ラッパーの登録	360
OLE DB ラッパー・ライブラリー・ファイル	361
OLE DB データ・ソースのサーバー定義の登録	361

CREATE SERVER ステートメント - OLE DB ラッパの例	362
OLE DB データ・ソースのユーザー・マッピングの作成	363
CREATE USER MAPPING ステートメント - OLE DB ラッパの例	364

第 19 章 Oracle データ・ソースへのアクセスの構成 365

フェデレーテッド・システムへの Oracle の追加	365
フェデレーテッド・サーバーへの Oracle データ・ソースの追加	365
Oracle 環境変数の設定	366
Oracle クライアント構成ファイルのセットアップおよびテスト	370
Oracle ラッパの登録	371
Oracle ラッパ・ライブラリー・ファイル	372
Oracle データ・ソースのサーバー定義の登録	373
CREATE SERVER ステートメント - Oracle ラッパの例	374
Oracle データ・ソースのユーザー・マッピングの作成	375
CREATE USER MAPPING ステートメント - Oracle ラッパの例	375
Oracle サーバーへの接続のテスト	377
Oracle 表およびビューのニックネームの登録	378
CREATE NICKNAME ステートメント - Oracle ラッパの例	378
Oracle データ・ソースのための構成のチューニングとトラブルシューティング	379
接続の問題	379

第 20 章 Sybase データ・ソースへのアクセスの構成 381

Sybase のフェデレーテッド・システムへの追加	381
フェデレーテッド・サーバーへの Sybase データ・ソースの追加	381
Sybase 環境変数の設定	382
Sybase クライアント構成ファイルのセットアップおよびテスト	384
Sybase ラッパの登録	386
Sybase ラッパ・ライブラリー・ファイル	387
Sybase データ・ソースのサーバー定義の登録	387
CREATE SERVER ステートメント - Sybase ラッパの例	389
Sybase データ・ソースのユーザー・マッピングの作成	391
CREATE USER MAPPING ステートメント - Sybase ラッパの例	391
Sybase サーバーへの接続のテスト	392
Sybase 表およびビューのニックネームの登録	393
CREATE NICKNAME ステートメント - Sybase ラッパの例	394
Sybase データ・ソースに対する構成のチューニングおよびトラブルシューティング	395
sp_helpindex エラーを解決する	395

第 21 章 表構造ファイル・データ・ソースへのアクセスの構成 397

表構造ファイルとは ?	397
表構造ファイルの属性	397
ソート・ファイル	398
ソートされていないファイル	398
DB2 Information Integrator が表構造ファイルを処理する方法	398
フェデレーテッド・システムへの表構造ファイルの追加	400
フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加	400
表構造ファイル・ラッパの登録	400
表構造ファイル・ラッパ・ライブラリー・ファイル	401
表構造ファイルのサーバーの登録	402
表構造ファイルのニックネームの登録	402
CREATE NICKNAME ステートメント - 表構造ファイル・ラッパの例	403
表構造ファイル・ラッパのファイル・アクセス・コントロール・モデル	404
表構造ファイル・ラッパに関する最適化のヒントと考慮事項	405
表構造ファイル・ラッパのメッセージ	405

第 22 章 Teradata データ・ソースへのアクセスの構成 409

フェデレーテッド・システムへの Teradata の追加	409
フェデレーテッド・サーバーへの Teradata データ・ソースの追加	409
Teradata サーバーへの接続のテスト	410
Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)	411
Teradata 環境変数の設定	412
Teradata ラッパの登録	415
Teradata ラッパ・ライブラリー・ファイル	416
Teradata データ・ソースのサーバー定義の登録	416
CREATE SERVER ステートメント - Teradata ラッパの例	418
Teradata データ・ソースのユーザー・マッピングの作成	419
CREATE USER MAPPING ステートメント - Teradata ラッパの例	420
フェデレーテッド・サーバーから Teradata サーバーへの接続のテスト	420
フェデレーテッド・サーバーでの Teradata ニックネーム	422
Teradata 表およびビューのニックネームの登録	422
CREATE NICKNAME ステートメント - Teradata ラッパの例	423
Teradata データ・ソースに対する構成の調整およびトラブルシューティング	424
ニックネームに対する UPDATE または DELETE 操作エラー	424

Teradata アクセス・ロギングのチューニングと使用不可	424		
libcliv2.so (AIX) のランタイム・リンクの使用可能化	424		
第 23 章 Web サービス・データ・ソースへのアクセスの構成	427		
Web サービス・ラッパーおよび Web サービス記述言語文書	427		
Web サービスのフェデレーテッド・システムへの追加	433		
フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加	433		
Web サービス・ラッパーの登録	434		
Web サービス・ラッパー・ライブラリー・ファイル	435		
Web サービス・データ・ソースのサーバー定義の登録	435		
Web サービス・データ・ソースのニックネームの登録	436		
ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項	452		
Web サービス・データ・ソース - 照会の例	456		
Web サービス・ラッパーのセキュリティー	459		
Web サービス・ラッパーのメッセージ	460		
第 24 章 XML データ・ソースへのアクセスの構成	463		
XML とは?	463		
フェデレーテッド・システムへの XML の追加	466		
フェデレーテッド・システムへの XML の追加	466		
XML ラッパーの登録	467		
XML ラッパー・ライブラリー・ファイル	468		
XML データ・ソースのサーバーの登録	469		
XML データ・ソースのニックネームの登録	471		
非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)	479		
XML コスト・モデル機能の最適化のヒント	480		
XML データ・ソース - 照会の例	481		
XML ラッパーのメッセージ	482		
第 4 部 ユーザー定義関数	493		
第 25 章 ライフ・サイエンス・ユーザー定義関数	495		
ライフ・サイエンス・ユーザー定義関数の概要	495		
ライフ・サイエンス・ユーザー定義関数の概要	495		
ライフ・サイエンス・ユーザー定義関数ライブラリー・ファイル	495		
機能カテゴリーごとのライフ・サイエンス・ユーザー定義関数	496		
ライフ・サイエンス・ユーザー定義関数の登録	497		
ライフ・サイエンス・ユーザー定義関数の使用不可	499		
逆変換ユーザー定義関数	500		
逆変換ユーザー定義関数 - 概要	500		
LSPep2AmbNuc ユーザー定義関数	500		
LSPep2AmbNuc ユーザー定義関数の例	501		
LSPep2AmbNuc ユーザー定義関数のエラー・メッセージ	503		
LSPep2ProbNuc ユーザー定義関数	503		
LSPep2ProbNuc ユーザー定義関数の例	504		
LSPep2ProbNuc ユーザー定義関数のエラー・メッセージ	505		
定義行構文解析のユーザー定義関数	506		
定義行構文解析のユーザー定義関数 - 概要	506		
LSDefineParse ユーザー定義関数	506		
LSDefineParse ユーザー定義関数の例	508		
一般化されたパターン・マッチングのユーザー定義関数	511		
LSPatternMatch ユーザー定義関数	512		
LSPatternMatch ユーザー定義関数の例	512		
LSPrositePattern ユーザー定義関数	514		
LSPrositePattern ユーザー定義関数の例	514		
正規表現のサポート	515		
GeneWise ユーザー定義関数	515		
GeneWise へのリンク	516		
LSGeneWise ユーザー定義関数	516		
LSGeneWise ユーザー定義関数の例	518		
Motifs ユーザー定義関数	518		
LSBarcode ユーザー定義関数	518		
LSBarcode ユーザー定義関数の例	519		
LSMultiMatch ユーザー定義関数	520		
LSMultiMatch ユーザー定義関数の例	521		
LSMultiMatch3 ユーザー定義関数	522		
LSMultiMatch3 ユーザー定義関数の例	522		
反転ユーザー定義関数	524		
LSRevComp ユーザー定義関数	524		
LSRevComp ユーザー定義関数の例	524		
LSRevNuc ユーザー定義関数	525		
LSRevNuc ユーザー定義関数の例	526		
LSRevPep ユーザー定義関数	526		
LSRevPep ユーザー定義関数の例	527		
変換	527		
LSNuc2Pep ユーザー定義関数	527		
LSNuc2Pep ユーザー定義関数の例	528		
LSTransAllFrames ユーザー定義関数	529		
LSTransAllFrames ユーザー定義関数	530		
コドン度数表の形式	531		
コドン度数表の例	532		
変換表の形式	532		
変換表の例	533		
第 26 章 KEGG ユーザー定義関数	535		
KEGG ユーザー定義関数 - 概要	535		
機能カテゴリーごとの KEGG ユーザー定義関数	536		
KEGG ユーザー定義関数の関数引き数	537		
KEGG ユーザー定義関数の登録	538		
パス・データベース関数	540		
GenesByPathwyS ユーザー定義関数	540		
GenesByPathwyT ユーザー定義関数	541		

	CompoundsByPathwyS ユーザー定義関数	541
	CompoundsByPathwyT ユーザー定義関数	542
	EnzymesByPathwyS ユーザー定義関数	543
	EnzymesByPathwyT ユーザー定義関数	543
	PathwysByGenesS ユーザー定義関数	544
	PathwysByGenesT ユーザー定義関数	545
	PathwysByCompndsS ユーザー定義関数	546
	PathwysByCompndsT ユーザー定義関数	546
	PathwysByEnzymesS ユーザー定義関数	547
	PathwysByEnzymesT ユーザー定義関数	548
	Sequence Similarity データベース関数	549
	SSDB データベース照会 (表関数) から戻される 列	549
	AllNbrsByGeneS ユーザー定義関数	550
	AllNbrsByGeneT ユーザー定義関数	551
	BstBstNbrsByGeneS ユーザー定義関数	552
	BstBstNbrsByGeneT ユーザー定義関数	553
	BestNbrsByGeneS ユーザー定義関数	554
	BestNbrsByGeneT ユーザー定義関数	555
	RevBestNbrsByGeneS ユーザー定義関数	556
	RevBestNbrsByGeneT ユーザー定義関数	557
	ParalogsByGeneS ユーザー定義関数	558
	ParalogsByGeneT ユーザー定義関数	559
	BestHmlgsByGenesS ユーザー定義関数	560
	BestHmlgsByGenesT ユーザー定義関数	560
	BstBstHmlgByGenesS ユーザー定義関数	561
	BstBstHmlgByGenesT ユーザー定義関数	562
	ScoreBetweenGenesS ユーザー定義関数	563
	DefinitionsByGeneS ユーザー定義関数	563
	GenesByMotifsT ユーザー定義関数	564
	KEGG ユーザー定義関数を使用不可にする	565

第 5 部 参照情報 567

第 27 章 非リレーショナル・データ・ ソースでサポートされるデータ・タイプ. 569

	非リレーショナル・データ・ソースでサポートされ るデータ・タイプ	569
	BioRS ラッパーによってサポートされるデー タ・タイプ	569
	BLAST ラッパーによってサポートされるデー タ・タイプ	569
	Documentum ラッパーによってサポートされるデー タ・タイプ	570
	Entrez ラッパーによってサポートされるデー タ・タイプ	570
	Excel ラッパーによってサポートされるデータ・ タイプ	571
	Extended Search ラッパーによってサポートされ るデータ・タイプ	571
	HMMER ラッパーによってサポートされるデー タ・タイプ	571
	表構造ファイルのラッパーによってサポートされ るデータ・タイプ	571

	Web サービス・ラッパーによってサポートされ るデータ・タイプ	572
	WebSphere Business Integration ラッパーによ ってサポートされるデータ・タイプ	572
	XML ラッパーによってサポートされるデータ・ タイプ	573

第 28 章 ニックネームの変更. 575

	ニックネームの変更	575
	ニックネームの変更に関する制限	576
	ニックネームの列名の変更	578
	ニックネーム・オプションの変更	580
	ニックネーム列オプションの変更	581
	データ・ソース・オブジェクトのローカル・タイ プの変更	583
	データ・ソース・オブジェクトのローカル・タイ プの変更例	584

第 29 章 DDL コマンド・リファレンス 587

	BioRS DDL 参照情報	587
	CREATE SERVER ステートメント・オプション - BioRS ラッパー	587
	CREATE USER MAPPING ステートメント・オ プション - BioRS ラッパー	588
	CREATE NICKNAME ステートメント構文 - BioRS ラッパー	589
	BLAST DDL 参照情報	590
	CREATE SERVER ステートメントの引き数 - BLAST ラッパー	590
	CREATE NICKNAME ステートメント構文 - BLAST ラッパー	591
	Documentum DDL 参照情報	592
	CREATE SERVER ステートメントの引き数とオ プション - Documentum ラッパー	592
	CREATE USER MAPPING ステートメントのオ プション - Documentum ラッパー	593
	CREATE NICKNAME ステートメント構文 - Documentum ラッパー	594
	Excel DDL 参照情報	597
	CREATE SERVER ステートメントの引き数 - Excel ラッパー	597
	CREATE NICKNAME ステートメント構文 - Excel ラッパー	597
	Extended Search DDL 参照情報	599
	CREATE WRAPPER ステートメント構文 - Extended Search ラッパー	599
	CREATE SERVER ステートメント構文 - Extended Search ラッパー	599
	CREATE USER MAPPING ステートメント構文 - Extended Search ラッパー	601
	CREATE NICKNAME ステートメント構文 - Extended Search ラッパー	601
	CREATE FUNCTION ステートメント構文 - Extended Search ラッパー	604
	Entrez DDL 参照情報	604

CREATE SERVER ステートメントの引き数 - Entrez ラッパー	604
CREATE NICKNAME ステートメントのオプション - Entrez ラッパー	605
表構造ファイル DDL 参照情報	605
CREATE NICKNAME ステートメント構文 - 表 構造ファイル・ラッパー	605
XML DDL 参照情報	610
CREATE NICKNAME ステートメント構文 - XML ラッパー	610

付録 A. フェデレーテッド情報を含むグローバル・カタログ表内のビュー 617

付録 B. フェデレーテッド・システムのラッパー・オプション 621

付録 C. SQL ステートメントで有効なサーバーのタイプ 623

BioRS ラッパー	623
BLAST ラッパー	623
CTLIB ラッパー	624
Documentum ラッパー	624
DRDA ラッパー	624
Entrez ラッパー	625
Excel ラッパー	625
Extended Search ラッパー	625
HMMER ラッパー	625
Informix ラッパー	626
MSSQLODBC3 ラッパー	626
NET8 ラッパー	626
ODBC ラッパー	626
OLE DB ラッパー	626
表構造ファイル・ラッパー	627
Teradata ラッパー	627
Web サービス・ラッパー	627
WebSphere Business Integration ラッパー	627
XML ラッパー	627

付録 D. フェデレーテッド・システムのサーバー・オプション 629

付録 E. フェデレーテッド・システムのユーザー・マッピング・オプション 647

付録 F. フェデレーテッド・システムのニックネーム・オプション 649

付録 G. フェデレーテッド・システムのニックネーム列オプション 661

付録 H. デフォルトの順方向データ・タイプ・マッピング 669

DB2 for z/OS and OS/390 データ・ソース	669
---	-----

DB2 for iSeries データ・ソース	670
DB2 Server for VM and VSE データ・ソース	671
DB2 for Linux、UNIX、および Windows のデータ・ソース	673
Informix データ・ソース	674
Microsoft SQL Server データ・ソース	675
ODBC データ・ソース	678
Oracle NET8 データ・ソース	679
Sybase データ・ソース	680
Teradata データ・ソース	681

付録 I. デフォルトの逆方向データ・タイプ・マッピング 685

DB2 for z/OS and OS/390 データ・ソース	686
DB2 for iSeries データ・ソース	687
DB2 for VM and VSE データ・ソース	688
DB2 for Linux、UNIX、および Windows のデータ・ソース	689
Informix データ・ソース	690
Microsoft SQL Server データ・ソース	691
Oracle NET8 データ・ソース	692
Sybase データ・ソース	693
Teradata データ・ソース	694

付録 J. フェデレーテッド・システムの関数マッピング・オプション 697

DB2 Information Integrator technical 技術文書 699

DB2 Information Integrator の資料	699
リリース情報およびインストール要件	700
DB2 Information Integrator ドキュメンテーション・フィックスパック	701

アクセス支援 703

キーボードによる入力およびナビゲーション	703
キーボード入力	703
キーボード・ナビゲーション	703
キーボード・フォーカス	703
アクセスしやすい表示	704
フォントの設定	704
色に依存しない	704
支援テクノロジーとの互換性	704
アクセスしやすい資料	704

特記事項 705

商標	707
索引	709

IBM と連絡を取る 721

製品情報	721
資料についてのコメント	721

本書について

本書では、データ・ソースにアクセスするためにフェデレーテッド・システムを構成する方法を説明します。本書の内容は、次のとおりです。

- フェデレーテッド・システムのコネクト、コンポーネント、および機能の紹介。
- データ・ソースの構成の計画についての推奨。
- データ・ソースへのアクセスに、フェデレーテッド・サーバーとデータベースが必要とするオブジェクトを登録するための指示。
- それぞれのデータ・ソースに固有の、広範囲な参照情報。これにはサーバー・セットアップ情報、環境変数の要件、データ・タイプ・マッピング、およびデータ・ソース構成のカスタマイズと調整のために設定できるオプションが含まれます。

技術的変更の位置

テキストに技術的変更がある場合、それは変更箇所の左側に縦線で示されます。

本書の対象読者

本書は、フェデレーテッド・システムのセットアップおよび構成をする必要のある、システム管理者、データベース管理者、セキュリティー管理者、およびシステム演算子を対象にしています。リレーショナル・データ・ソースおよび非リレーショナル・データ・ソースのデータにアクセスするフェデレーテッド・システムを管理するには、本書を使用してください。フェデレーテッド・システムの構成および使用について理解する必要のあるプログラマー、およびその他のユーザーも本書を使用することができます。本書は、読者に DB2 の知識があることを前提にして書かれています。読者は、標準的なデータベース用語に通じており、データベースの設計およびデータベースの管理の経験がある必要があります。本書では、読者が自分自身のアプリケーションおよびアクセスしたいデータ・ソースについて精通していることを前提としています。

本書の規則および用語

フェデレーテッドの用語:

本書の用語集で、フェデレーテッド・システムについて論じる際の用語が定義されています。

DB2 コマンド:

本書では、特に指定しない限り、DB2 コマンドは DB2 コマンド行プロセッサ (CLP) または DB2 コマンド・センター GUI に入力されることを前提とします。DB2 コマンドが本文中で取り上げられている場合、コマンドとそのパラメーターのみをリストします。コマンドの先頭に DB2 は付けません。

DB2 コントロール・センター:

資料には、DB2 コントロール・センターを使用して作業を行うタイミングの説明およびそれらの作業を行うステップが含まれています。

書体の規則:

本書では、以下のような書体の規則を使用しています。

太字体 コマンドやグラフィカル・ユーザー・インターフェース・コントロール (フィールド名、プッシュボタン名、メニュー選択など) を示します。太字体は注、制限、前提条件、および推奨の明示に使用されます。

モノスペース体

入力するテキスト、ファイル名、およびコード例を示します。モノスペース体は SQL ステートメントまたは DB2 コマンド・パラメーター名にも使用されます。

イタリック体

適切な値で置き換える SQL ステートメントまたは DB2 コマンド・パラメーター値を示します。SQL ステートメントまたは DB2 コマンド例では、サンプル・パラメーター値を示すためにイタリック体を使用されます。イタリック体は言葉を強調するため、また新しい用語や文書のタイトルを示すために使用されます。

英大文字体

DB2 コマンドおよび SQL ステートメント、またそれらのキーワードを示します。大文字はデータ・タイプ名、オプション、および頭字語にも使用されます。

構文図の読み方

本書では、構文は次のように定義された構造を用いて記述されています。

構文図は、左から右、上から下に、線に沿って読みます。

▶▶— シンボルは、ステートメントの開始を示します。

—▶ シンボルは、構文図が次の行に続くことを示します。

▶— シンボルは、ステートメントが前の行から続いていることを示します。

—▶▶ 記号は、ステートメントの終わりを示します。

必須項目は、水平線 (メインパス) の線上に示されます。

▶▶—STATEMENT—必須項目—▶▶

オプション項目は、メインパスの下に示されます。

▶▶—STATEMENT—
└ オプション項目 ┘▶▶

オプション項目がメインパスの上に示されている場合、その項目は読みやすくするために示されているのであり、ステートメントの実行には影響を与えません。

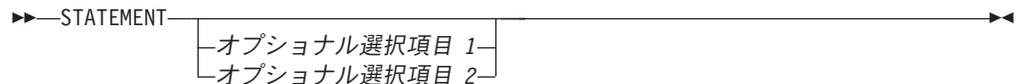


2 つ以上の項目からの選択が可能な場合は、複数の項目が縦に並んでスタックに記載されます。

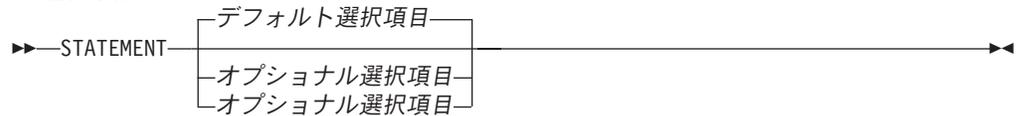
項目から 1 つを選択しなければならない場合、スタック内の項目の 1 つがメインパスの線上に示されます。



項目から何も選択しなくてもよい場合は、スタック全体がメインパスの下に表示されます。



項目のうちの 1 つがデフォルトの場合、それは、メインパスの上に表示され、残りの選択項目はその下に表示されます。



メインパスの上の左に折り返す矢印は、項目が繰り返されることを示しています。この場合、繰り返し項目は 1 つ以上の空白で区切る必要があります。



繰り返しの矢印がコンマを含む場合は、繰り返し項目をコンマで区切る必要があります。



繰り返しの矢印がスタックの上にある場合、スタック項目の中から 1 つ以上選択するか、1 つの選択項目を繰り返すことができることを示しています。

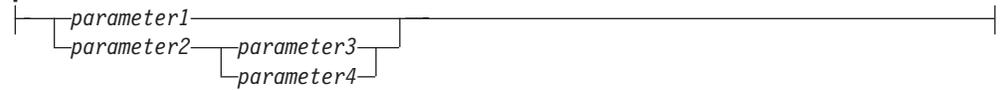
キーワードは大文字で表記されています (例: FROM)。キーワードの記載は、表記どおりでなければなりません。変数は小文字で表記されています (例: column-name)。変数は、構文中でのユーザー指定の名前または値を表します。

句読記号、括弧、算術演算子、または同様の他のシンボルが示されている場合、それらを構文の一部として入力しなければなりません。

単一の変数が、いくつかのパラメーターのセットを表していることがあります。たとえば、下記の図で、変数 parameter-block は、**parameter-block** という見出しの下の内容で置きかえられます。



parameter-block:



「黒丸」 (●) の間の隣り合うセグメントは、どのような順序で指定してもかまいません。



上記の図は、item2 と item3 は、どちらを先に指定してもかまわないことを示しています。次のいずれも有効です。

```
STATEMENT item1 item2 item3 item4  
STATEMENT item1 item3 item2 item4
```

第 1 部 概念および計画

第 1 章 フェデレーテッド・システムの概要

以下の各セクションでは、フェデレーテッド・システムについての概要を示します。

フェデレーテッド・システム

DB2[®] フェデレーテッド・システム は、特殊なタイプの分散データベース管理システム (DBMS) です。1 つのフェデレーテッド・システムは、フェデレーテッド・サーバーとして働く DB2 インスタンス、フェデレーテッド・データベースとして働くデータベース、1 つまたは複数のデータ・ソース、およびデータベースとデータ・ソースにアクセスするクライアント (ユーザーおよびアプリケーション) からなっています。フェデレーテッド・システムを使用すると、1 つの SQL ステートメントで複数のデータ・ソースに分散要求を送信することができます。たとえば、DB2 Universal Database[™] 表、Oracle 表、および XML タグ付きファイルにあるデータを 1 つの SQL ステートメントで結合できます。次の図は、フェデレーテッド・システムのコンポーネントおよびアクセス可能なデータ・ソースの例を示しています。

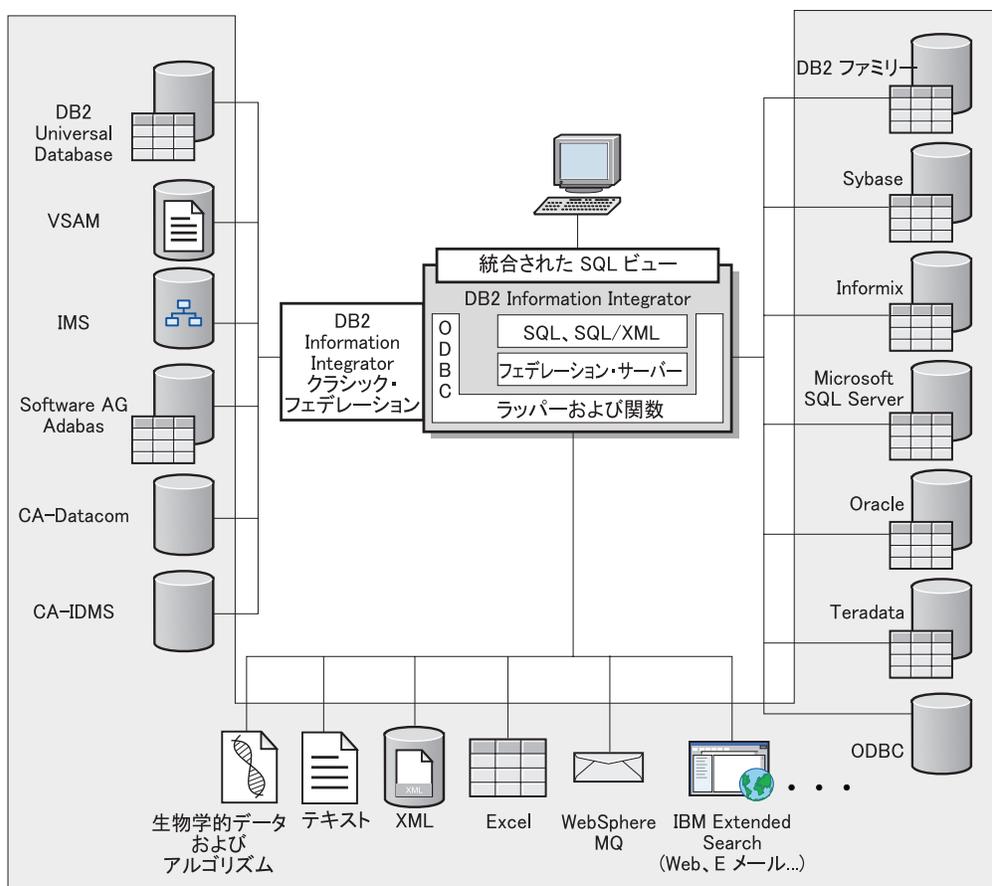


図1. フェデレーテッド・システムのコンポーネント

DB2 フェデレーテッド・システムの持つ強力な機能により、以下のことが可能になります。

- ローカル表とリモート・データ・ソースのデータを、それらすべてがフェデレーテッド・データベースにローカルに保管されているかのように結合する。
- リレーショナル・データ・ソースのデータを、それらがフェデレーテッド・データベースに保管されているかのように更新する。
- リレーショナル・データ・ソースとの間で双方向にデータを複製する。
- データ・ソースに要求を送信して処理させることにより、データ・ソース側で処理するという利点を生かす。
- データ・ソース側での SQL の制約を補うため、分散要求の一部をフェデレーテッド・サーバー側で処理する。

フェデレーテッド・サーバー

フェデレーテッド・システム内の DB2® サーバーのことを、フェデレーテッド・サーバー といいます。DB2 インスタンスであればいくつでも、フェデレーテッド・サーバーとして機能するように構成することができます。既存の DB2 インスタンスをフェデレーテッド・サーバーとして使用することも、フェデレーテッド・システム専用新しく作成することもできます。

フェデレーテッド・システムを管理する DB2 インスタンスのことをサーバーと呼びますが、それはこのインスタンスがエンド・ユーザーおよびクライアント・アプリケーションからの要求に応答するからです。フェデレーテッド・サーバーは受信した要求の各部を頻繁にデータ・ソースに送信して処理させます。プッシュダウン操作は、リモート側で処理される操作です。フェデレーテッド・システムを管理する DB2 インスタンスは、要求をデータ・ソースにプッシュダウンする場合はクライアントとして働きますが、フェデレーテッド・サーバーと呼ばれます。

その他のアプリケーション・サーバーと同様に、フェデレーテッド・サーバーはデータベース・マネージャー・インスタンスです。アプリケーション・プロセスはフェデレーテッド・サーバーに接続し、フェデレーテッド・サーバー内のデータベースに要求をサブミットします。ただし、次の 2 つの主要な機能により、その他のアプリケーション・サーバーとは区別されます。

- フェデレーテッド・サーバーは、部分的または全面的にデータ・ソース向けの要求を受信するように構成されています。フェデレーテッド・サーバーは、これらの要求をデータ・ソースに配布します。
- その他のアプリケーション・サーバーと同様に、フェデレーテッド・サーバーは DRDA[®] 通信プロトコル (TCP/IP を介する) を使用して、DB2 ファミリーのインスタンスと通信します。ただし、他のアプリケーション・サーバーと異なり、フェデレーテッド・サーバーはデータ・ソースのネイティブ・クライアントを使用して、データ・ソースにアクセスします。たとえば、フェデレーテッド・サーバーは Sybase Open Client を使用して Sybase データ・ソースにアクセスし、Microsoft SQL Server ODBC ドライバーを使用して Microsoft[®] SQL Server データ・ソースにアクセスします。

関連概念:

- 5 ページの『データ・ソースとは?』

データ・ソースとは?

フェデレーテッド・システムでは、リレーショナル DBMS インスタンス (Oracle または Sybase など) または非リレーショナル・データ・ソース (BLAST 検索アルゴリズムまたは XML タグ付きファイル) をデータ・ソースにすることができます。特定のデータ・ソースを介することで、他のデータ・ソースにアクセスすることも可能です。たとえば、Extended Search データ・ソースを介して、Lotus[®] Notes データベース、Microsoft[®] Access、Microsoft Index Server、Web 検索エンジン、および Lightweight Directory Access Protocol (LDAP) ディレクトリーなどのデータ・ソースにアクセスできます。

データ・ソースへのアクセスに使用される方式つまりプロトコルは、データ・ソースのタイプによって異なります。たとえば、DB2[®] for z/OS[™] and OS/390[®] データ・ソースにアクセスするには DRDA[®] が使用され、Documentum データ・ソースにアクセスするには Documentum Client API/Library が使用されます。

データ・ソースは準自立的です。たとえば、フェデレーテッド・サーバーが Oracle データ・ソースに照会を送信しているときに、その同じデータ・ソースに Oracle アプリケーションがアクセスしてもかまいません。保全性およびロッキング制約が損

なわれない限り、DB2 フェデレーテッド・システムが他のデータ・ソースへのアクセスを独占または制限することはありません。

関連概念:

- 8 ページの『フェデレーテッド・データベース』

関連資料:

- 6 ページの『サポートされるデータ・ソース』

サポートされるデータ・ソース

フェデレーテッド・システムを使用してアクセスできるデータ・ソースはたくさんあります。次の表は、サポートされるデータ・ソースを示しています。

表 1. サポート対象データ・ソースのバージョンおよびアクセス方式

データ・ソース	サポートされるバージョン	アクセス方式
DB2 Universal Database™ for Linux, UNIX, and Windows®	7.2、8.1、8.2	DRDA®
DB2 Universal Database for z/OS™ and OS/390®	6.1、7.1 で以下の APAR が適用されているもの: <ul style="list-style-type: none">• PQ62695• PQ55393• PQ56616• PQ54605• PQ46183• PQ62139 8.1	DRDA
DB2 Universal Database for iSeries™	5.1 <ul style="list-style-type: none">• 以下の APAR が適用 されているもの: – SE06003– SE06872– III3348 <ul style="list-style-type: none">• 以下の PTF が適用さ れているもの: – SI05990 SI05991	DRDA
DB2 Server for VM and VSE	5.2 で PTF SI0735 が適 用されているもの。	DRDA
DB2 Server for VM and VSE	7.1 (またはそれ以降) で スキーマ関数用の APAR のフィックスが適用され ているもの。	DRDA

表 1. サポート対象データ・ソースのバージョンおよびアクセス方式 (続き)

データ・ソース	サポートされるバージョン	アクセス方式
Informix™	7.31、8.32、8.4、9.3、9.4	Informix Client SDK V2.7 以上
ODBC	3.x	データ・ソース用の ODBC ドライバー (Redbrick にアクセスする場合は Redbrick ODBC ドライバー、など)
OLE DB	2.7、2.8	OLE DB 2.0 以上
Oracle	8.0.6、8.1.6、8.1.7、9.0、9.1、9.2、9i、10g	Oracle ネット・クライアントまたは NET8 クライアント・ソフトウェア
Microsoft SQL Server	7.0、2000 SP3 およびそのリリースでそれ以降のサービス・パックが適用されたもの	Windows では、Microsoft SQL Server Client ODBC 3.0 (またはそれ以降の) ドライバー UNIX では、DataDirect Technologies (以前の MERANT) Connect ODBC 3.7 (またはそれ以降の) ドライバー
Sybase	11.9.2、12.x	Sybase Open Client ctlib インターフェース
Teradata	V2R3、V2R4、V2R5	Teradata Call-Level Interface, Version 2 (CLIV2) Release 04.06 (またはそれ以降)
BLAST	2.2.3 以降、2.2 フィックスパックがサポートされる	BLAST デーモン (ラッパーと一緒に提供される)
BioRS	v5.0.14	なし
Documentum	3.x、4.x	Documentum Client ライブラリー/APL3.1.7a 以上
Entrez (PubMed および GenBank データ・ソース)	1.0	なし
HMMER	2.2g、2.3	HMMER デーモン (ラッパーと一緒に提供される)
IBM Lotus Extended Search	4.0.1、4.0.2	Extended Search クライアント・ライブラリー (ラッパーと一緒に提供される)
Microsoft Excel	97、2000、2002、2003	フェデレーテッド・サーバーにインストールされている Excel 97、2000、2002、または 2003
PeopleSoft	8.x	IBM WebSphere Business Integration Adapter for PeopleSoft v2.3.1、2.4

表 1. サポート対象データ・ソースのバージョンおよびアクセス方式 (続き)

データ・ソース	サポートされるバージョン	アクセス方式
SAP	3.x、4.x	IBM WebSphere Business Integration Adapter for mySAP.com v2.3.1、2.4
Siebel	7、7.5、2000	IBM WebSphere Business Integration Adapter for Siebel eBusiness Applications v2.3.1、2.4
表構造ファイル		なし
KEGG 用ユーザー定義関数	サポートされる	
Life Sciences 用ユーザー定義関数	サポートされる	
Web サービス	SOAP 1.0、1.1、WSDL 1.0、1.1 仕様	HTTP
XML	1.0 仕様	なし

関連概念:

- 5 ページの『データ・ソースとは?』

フェデレーテッド・データベース

エンド・ユーザーとクライアント・アプリケーションにとっては、データ・ソースは DB2® 内の 1 まとまりのデータベースとして認識されます。ユーザーとアプリケーションは、フェデレーテッド・サーバーが管理するフェデレーテッド・データベースとやり取りを行います。フェデレーテッド・データベースにはシステム・カタログが入っています。このフェデレーテッド・データベースのシステム・カタログには、データ・ソースとその特性を示すカタログ項目が入っています。フェデレーテッド・サーバーは、フェデレーテッド・データベース・システム・カタログに保管された情報およびデータ・ソース・ラッパーを検討した上で、SQL ステートメントを処理する最善のプランを決めます。

フェデレーテッド・システムは、データ・ソースがフェデレーテッド・データベース内の通常のリレーショナルの表またはビューであるかのように、SQL ステートメントを処理します。その結果、次のようになります。

- フェデレーテッド・システムはリレーショナル・データを非リレーショナルのフォーマットのデータと結合することができます。データ・ソースが異なる SQL ダイアレクトを使用していたり、あるいは SQL をまったくサポートしていても、あてはまります。
- フェデレーテッド・データベースの特性とデータ・ソースの特性に相違がある場合、フェデレーテッド・データベースの特性が優先されます。
 - 一例として、フェデレーテッド・サーバーで使用されるコード・ページが、データ・ソースで使用されるコード・ページと異なっているとします。データ・

ソースの文字データは、フェデレーテッド・ユーザーに戻される際、フェデレーテッド・データベースで使用されているコード・ページに基づいて変換されます。

- 別の例として、フェデレーテッド・サーバーで使用される照合シーケンスが、データ・ソースで使用される照合シーケンスと異なっているとします。文字データに対するソート操作はすべて、データ・ソースではなくフェデレーテッド・サーバーで行われます。

関連概念:

- 10 ページの『SQL コンパイラー』
- 9 ページの『フェデレーテッド・データベース・システム・カタログ』

フェデレーテッド・データベース・システム・カタログ

フェデレーテッド・データベース・システム・カタログには、フェデレーテッド・データベース内のオブジェクトの情報と、データ・ソース側のオブジェクトの情報が入っています。フェデレーテッド・データベース内のカタログは、フェデレーテッド・システム全体についての情報が入っているため、*グローバル・カタログ* と呼びます。DB2® 照会オプティマイザーは、*グローバル・カタログ*内の情報およびデータ・ソース・ラッパーを使用して、SQL ステートメントを処理する最善の方法を計画します。*グローバル・カタログ*に保管される情報には、リモートとローカルの情報、たとえば列名、列のデータ・タイプ、列のデフォルト値、および索引情報などが含まれます。

リモート・カタログ情報は、データ・ソースが使用する情報または名前です。ローカル・カタログ情報は、フェデレーテッド・データベースが使用する情報または名前です。たとえば、*EMPNO* という名前の列を持つリモート表があるとします。*グローバル・カタログ*には、このリモートの列名が *EMPNO* として保管されます。別の名前を指定しないかぎり、ローカルの列名は *EMPNO* として保管されます。ローカルの列名を *Employee_Number* に変更することができます。この列を含む照会をサブミットするユーザーは、照会の中で *EMPNO* ではなく *Employee_Number* を使用します。データ・ソース列のローカル名を変更するには、ALTER NICKNAME ステートメントを使用します。

リレーショナル・データ・ソースの場合、*グローバル・カタログ*に保管される情報にはリモートとローカルの両方の情報が含まれます。

非リレーショナルのデータ・ソースの場合、*グローバル・カタログ*に保管される情報はデータ・ソースによって異なります。

*グローバル・カタログ*に保管されたデータ・ソース表の情報を見るには、フェデレーテッド・データベース内の SYSCAT.TABLES、SYSCAT.TABOPTIONS、SYSCAT.INDEXES、SYSCAT.COLUMNS、および SYSCAT.COLOPTIONS カタログ・ビューを照会してください。

*グローバル・カタログ*には、データ・ソースについてのその他の情報も入っています。たとえば、フェデレーテッド・サーバーがデータ・ソースに接続したり、フェデレーテッド・ユーザー許可をデータ・ソースのユーザー権限にマップするために

使用する情報が含まれます。グローバル・カタログには、明示的に設定されたデータ・ソースの属性 (サーバー・オプションなど) が入っています。

関連概念:

- 10 ページの『SQL コンパイラー』

関連資料:

- 617 ページの『付録 A. フェデレーテッド情報を含むグローバル・カタログ表内のビュー』

SQL コンパイラー

データ・ソースからデータを入手するため、ユーザーおよびアプリケーションは DB2® SQL の照会をフェデレーテッド・データベースにサブミットします。照会をサブミットすると、DB2 SQL コンパイラーはグローバル・カタログ内の情報およびデータ・ソース・ラッパーを検討し、照会の処理に役立てます。この情報には、データ・ソースへの接続に関する情報、サーバー属性、マッピング、索引情報、および処理統計が含まれます。

関連概念:

- 13 ページの『ラッパーおよびラッパー・モジュール』
- 10 ページの『照会オプティマイザー』

照会オプティマイザー

SQL コンパイラー処理の一部として、照会オプティマイザー は照会を分析します。コンパイラーは、アクセス・プラン と呼ばれる、照会を処理するための代替ストラテジーを作成します。アクセス・プランは、照会を次のように処理することを要求します。

- 照会をデータ・ソースが処理する
- 照会をフェデレーテッド・サーバーが処理する
- 照会の一部をデータ・ソースが処理し、一部をフェデレーテッド・サーバーが処理する

DB2® UDB は、主にデータ・ソースの能力およびデータに関する情報を基にアクセス・プランを評価します。この情報はラッパーとグローバル・カタログにあります。DB2 UDB は照会を照会フラグメント と呼ばれるセグメントに分解します。通常、照会フラグメントをデータ・ソースにプッシュダウン した方が、より効率的です (データ・ソースがフラグメントを処理できる場合)。しかし、照会オプティマイザーは次のような他の要素も考慮します。

- 処理する必要のあるデータの量
- データ・ソースの処理速度
- フラグメントが戻すデータの量
- 通信の帯域幅
- 同じ照会結果を表す、使用可能なマテリアライズ照会表がフェデレーテッド・サーバーにあるかどうか

照会オプティマイザーは、照会フラグメントを処理するためのローカルとリモートのアクセス・プランを、リソースのコストに基づいて生成します。それから、DB2 UDB は最小のリソース・コストで照会を処理すると思われるプランを選択します。

何らかのフラグメントをデータ・ソースで処理する場合、DB2 UDB はそれらのフラグメントをデータ・ソースにサブミットします。データ・ソースがフラグメントを処理した後、結果が取り出されて DB2 UDB に戻されます。DB2 UDB が処理の一部を実行する場合、その処理結果とデータ・ソースから取り出した結果が組み合わされます。それから、DB2 UDB はすべての結果をクライアントに戻します。

関連概念:

- 10 ページの『SQL コンパイラー』
- 11 ページの『適合』
- 「フェデレーテッド・システム・ガイド」の『照会処理のチューニング』

適合

データ・ソースが照会フラグメントを処理できない場合、またはデータ・ソースで処理するよりもフェデレーテッド・サーバーで処理した方が速い場合、DB2® フェデレーテッド・サーバーは照会フラグメントをプッシュダウンしません。たとえば、データ・ソースの SQL ダイアレクトが、GROUP BY 文節の中では CUBE グループ化をサポートしていないとします。この場合、CUBE グループ化を含み、かつそのデータ・ソース内の表を参照する照会は、フェデレーテッド・サーバーにサブミットされます。DB2 Information Integrator は CUBE グループ化をデータ・ソースにプッシュダウンせず、CUBE 自体を処理します。データ・ソースではサポートされていない SQL を処理する DB2 Information Integrator の機能を、適合 と呼びます。

フェデレーテッド・サーバーは、データ・ソース側の機能不足を次の 2 つの方法で補います (適合させます)。

- 照会に記述されている DB2 関数と同等の、1 つまたは複数の操作を使用するようデータ・ソースに求めることができます。たとえば、あるデータ・ソースがコタンジェント (COT(x)) 関数をサポートしておらず、タンジェント (TAN(x)) 関数をサポートしているとします。このとき DB2 Information Integrator は、コタンジェント (COT(x)) 関数と同等の計算 (1/TAN(x)) を実行するようデータ・ソースに求めることができます。
- データ集合をフェデレーテッド・サーバーに戻し、関数をローカルで実行することができます。

リレーショナル・データ・ソースの場合は、RDBMS の各タイプがそれぞれ国際 SQL 標準のサブセット 1 つをサポートします。また、この標準を超える SQL 構造をサポートする RDBMS のタイプもあります。SQL ダイアレクト とは、1 つの RDBMS タイプがサポートする SQL の全体を指す用語です。ある SQL 構造が DB2 の SQL ダイアレクトにあり、リレーショナル・データ・ソース側のダイアレクトにはない場合、フェデレーテッド・サーバーはそのデータ・ソースに代わってこの構造をインプリメントすることができます。

DB2 Information Integrator は SQL ダイアレクト内の相違を補う (適合させる) ことができます。その一例は `common-table-expression` (共通表式) という文節で、この文節は DB2 SQL に組み込まれています。この文節に指定できるのは、全選択内のどの FROM 文節に指定しても結果セットを参照できる名前です。データ・ソースが使用する SQL ダイアレクトに共通表式が組み込まれていなくても、フェデレーテッド・サーバーはそのデータ・ソースの共通表式を処理します。

「適合」機能のおかげで、フェデレーテッド・サーバーはデータ・ソースを照会する際に、DB2 SQL ダイアレクトを完全にサポートすることができます。SQL サポートが弱いまたは SQL サポートがないデータ・ソースに対してであっても、適合は適用されます。パススルー・セッションの場合を除き、DB2 SQL ダイアレクトは必ずフェデレーテッド・システムで使用するようになっています。

関連概念:

- 12 ページの『パススルー・セッション』

パススルー・セッション

パススルー と呼ばれる特殊モードを使用すると、SQL ステートメントをデータ・ソースに直接サブミットすることができます。この場合 SQL ステートメントは、データ・ソースで使用されている SQL ダイアレクトを使用してサブミットします。しかし、DB2[®] SQL/API ではできない操作を実行したい場合には、パススルー・セッションを使用してください。たとえば、プロシーチャーの作成、索引の作成、またはデータ・ソースに固有のダイアレクトを使用した照会の実行には、パススルー・セッションを使用します。

現在、パススルーをサポートするデータ・ソースは、SQL を使用するパススルーをサポートします。将来的には、データ・ソースが SQL 以外のデータ・ソース言語を使用するパススルーをサポートすることもあります。

同様に、パススルー・セッションを使用して、SQL がサポートしていないアクション (たとえば、ある種の管理用タスク) を実行することもできます。ただし、パススルー・セッションを使用してすべての管理用タスクを実行することはできません。たとえば、データ・ソース側で表を作成したりドロップすることはできますが、リモート・データベースを開始または停止することはできません。

パススルー・セッションでは、静的 SQL と動的 SQL の両方を使用することができます。

フェデレーテッド・サーバーには、パススルー・セッションを管理するための次の SQL ステートメントが提供されています。

SET PASSTHRU

パススルー・セッションを開く。別の SET PASSTHRU ステートメントを発行して新しいパススルー・セッションを開始すると、現行のパススルー・セッションは終了します。

SET PASSTHRU RESET

現行のパススルー・セッションを終了する。

GRANT (Server Privileges)

ユーザー、グループ、許可 ID のリスト、または PUBLIC に対して、特定のデータ・ソースにパススルー・セッションを開始する権限を付与する。

REVOKE (Server Privileges)

パススルー・セッションを開始する権限を取り消す。

パススルー・セッションには以下の制約事項があります。

- データ・ソースの SQL ダイアレクトまたは言語コマンドを使用しなければならず、DB2 SQL ダイアレクトは使用できません。その結果、ニックネームを照会するのではなく、データ・ソースのオブジェクトを直接照会することになります。
- パススルー・セッションで UPDATE または DELETE 操作を実行する場合、WHERE CURRENT OF CURSOR 条件は使用できません。
- LOB は、パススルー・セッションではサポートされません。

関連概念:

- 13 ページの『ラッパーおよびラッパー・モジュール』
- 「フェデレーテッド・システム・ガイド」の『パススルーによりデータ・ソースを直接照会する』

ラッパーおよびラッパー・モジュール

ラッパーとは、フェデレーテッド・サーバーがデータ・ソースと対話するためのメカニズムです。フェデレーテッド・サーバーは、ライブラリーに保管されたルーチン (ラッパー・モジュール という) を使用してラッパーをインプリメントします。これらのルーチンを使用することで、フェデレーテッド・サーバーは、データ・ソースへの接続やデータ・ソースからのデータ検索の繰り返しなどの操作を実行できます。通常、DB2® フェデレーテッド・インスタンスの所有者は、CREATE WRAPPER ステートメントを使用して、ラッパーをフェデレーテッド・データベースに登録します。DB2_FENCED オプションを使用すると、ラッパーを fenced またはトラステッドとして登録することができます。

ラッパーは、アクセスするデータ・ソースのタイプごとに 1 つ作成します。たとえば、3 つの DB2 for z/OS™ データベース表、1 つの DB2 for iSeries™ の表、2 つの Informix® の表、および 1 つの Informix ビューにアクセスしたいとします。このとき作成する必要があるのは、DB2 データ・ソース・オブジェクト用のラッパーを 1 つと、Informix データ・ソース・オブジェクト用のラッパーを 1 つです。これらのラッパーをフェデレーテッド・データベースに登録すれば、すぐにこれらのラッパーを使用して対応するデータ・ソースから他のオブジェクトにアクセスすることが可能になります。たとえば、DRDA® ラッパーであれば、すべての DB2 ファミリーのデータ・ソース・オブジェクト (DB2 for Linux, UNIX®, and Windows®, DB2 for z/OS and OS/390®, DB2 for iSeries, および DB2 Server for VM and VSE) に使用することができます。

各データ・ソース・オブジェクトを特定して識別 (名前やロケーションなど) するには、サーバー定義とニックネームを使用します。

ラッパーは多くの作業を行います。そのいくつかは次のようなものです。

- データ・ソースに接続します。ラッパーは、データ・ソースの標準の接続 API を使用します。
- データ・ソースに照会をサブミットします。
 - SQL をサポートするデータ・ソースの場合、照会は SQL でサブミットされません。
 - SQL をサポートしないデータ・ソースの場合、照会は、ソースに固有の照会言語に、または一連のソース API 呼び出しに変換されます。
- データ・ソースから結果セットを受け取ります。ラッパーは、データ・ソースの標準 API を使用して、結果セットを受信します。
- データ・ソースのデフォルトのデータ・タイプ・マッピングについてのフェデレーテッド・サーバーの照会に応答します。ラッパーには、データ・ソース・オブジェクトにニックネームを作成する時に使用される、デフォルトのタイプ・マッピングが入っています。リレーショナル・ラッパーの場合、ユーザーが作成するデータ・タイプ・マッピングは、デフォルトのデータ・タイプ・マッピングをオーバーライドします。ユーザー定義のデータ・タイプ・マッピングは、グローバル・カタログに保管されます。
- データ・ソースのデフォルトの関数マッピングについてのフェデレーテッド・サーバーの照会に応答します。ラッパーには、DB2 関数がデータ・ソースの関数と対応付けられるかどうか、またどのように関数が対応付けられるかを、フェデレーテッド・サーバーが判断する際に必要となる情報が含まれています。この情報は、データ・ソースが照会操作を実行できるかどうかを判断するために、SQL コンパイラーにより使用されます。リレーショナル・ラッパーの場合、ユーザーが作成する関数マッピングは、デフォルトの関数タイプ・マッピングをオーバーライドします。ユーザー定義の関数マッピングは、グローバル・カタログに保管されます。

ラッパー・オプション は、ラッパーの構成や、DB2 Information Integrator がラッパーを使用する方法の定義に使用されます。

関連タスク:

- 「*IBM DB2 Information Integrator* ラッパー開発者向けガイド」の『トラステッドおよび fenced モードのプロセス環境』
- 68 ページの『データ・ソースのラッパーの登録』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』
- 621 ページの『付録 B. フェデレーテッド・システムのラッパー・オプション』

デフォルトのラッパー名

サポート対象のデータ・ソースごとに、ラッパーがあります。いくつかのラッパーにはデフォルト名があります。デフォルト名を使用してラッパーを作成すると、フェデレーテッド・サーバーはそのラッパーに関連するデータ・ソース・ライブラリーを自動的に選択します。

表2. データ・ソースごとのデフォルトのラッパー名

データ・ソース	デフォルトのラッパー名
DB2 Universal Database™ for Linux、DB2 Universal Database™ for UNIX、および DB2 Universal Database™ for Windows®	DRDA
DB2 Universal Database for z/OS and OS/390®	DRDA
DB2 Universal Database for iSeries	DRDA
DB2 Server for VM and VSE	DRDA
Informix	INFORMIX
Microsoft® SQL Server	MSSQLODBC3
ODBC	ODBC
OLE DB	OLEDB
Oracle	NET8
Sybase	CTLIB
Teradata	TERADATA
BLAST	なし
BioRS	なし
Documentum	なし
Entrez	なし
Extended Search	なし
HMMER	なし
Microsoft Excel	なし
表構造ファイル	なし
Web サービス	なし
WebSphere Business Integration	なし
XML	なし

関連概念:

- 13 ページの『ラッパーおよびラッパー・モジュール』

サーバー定義およびサーバー・オプション

データ・ソース用のラッパーを作成した後、フェデレーテッド・インスタンスの所有者はデータ・ソースをフェデレーテッド・データベースに定義します。インスタンス所有者は、データ・ソースを識別するための名前を指定し、またデータ・ソースに関するその他の情報も指定します。この情報には、次のものが含まれます。

- データ・ソースのタイプおよびバージョン
- データ・ソースのデータベース名 (RDBMS のみ)
- データ・ソースに固有のメタデータ

たとえば、DB2® ファミリーのデータ・ソースは複数のデータベースを持つことができます。そのため、フェデレーテッド・サーバーがどのデータベースに接続でき

るかを定義に指定しておく必要があります。それとは対照的に、Oracle データ・ソースが持つデータベースは 1 つなので、フェデレーテッド・サーバーは名前を知らなくてもそのデータベースに接続することができます。そのため、Oracle データ・ソースのフェデレーテッド・サーバー定義にデータベース名は含まれていません。

インスタンス所有者がフェデレーテッド・サーバーに提供する、名前およびその他の情報をまとめてサーバー定義と呼びます。データ・ソースはデータを求める要求に応答し、それ自体がサーバーとして機能します。

サーバー定義の作成および変更には、CREATE SERVER および ALTER SERVER ステートメントを使用します。

サーバー定義内の情報の一部は、サーバー・オプションとして保管されます。サーバー定義を作成するにあたって、サーバーに関して指定可能なオプションを理解しておくことは大切です。サーバー・オプションの中には、ラッパーを構成するオプションや、DB2 Information Integrator がラッパーを使用する方法に影響するオプションがあります。

サーバー・オプションは、データ・ソースへの接続が次々に続く間は持続されるように設定するか、または 1 つの接続が継続している間のみ持続されるように設定することができます。

関連タスク:

- 69 ページの『データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』

ユーザー・マッピング

フェデレーテッド・サーバーがデータ・ソースに要求をプッシュダウンする必要がある場合、サーバーはまずデータ・ソースに接続を確立する必要があります。

ほとんどのデータ・ソースの場合、フェデレーテッド・サーバーはそのデータ・ソースに対して有効なユーザー ID とパスワードを使用して接続を行います。データ・ソースに接続するためにユーザー ID とパスワードが必要な場合、フェデレーテッド・サーバーの許可 ID とデータ・ソースのユーザー ID およびパスワードとの間に関連付けを定義しておくことができます。分散要求を送信するためにフェデレーテッド・システムを使用する各ユーザー ID に対して、この関連付けを作成することができます。この関連付けをユーザー・マッピングといいます。

フェデレーテッド・データベースに接続するとき使用するユーザー ID およびパスワードが、リモート・データ・ソースにアクセスするとき使用するものと同じであれば、ユーザー・マッピングを作成する必要のない場合もあります。

関連タスク:

- 70 ページの『データ・ソースのユーザー・マッピングの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 647 ページの『付録 E. フェデレーテッド・システムのユーザー・マッピング・オプション』

ニックネームとデータ・ソース・オブジェクト

サーバー定義およびユーザー・マッピングを作成した後、フェデレーテッド・インスタンスの所有者はニックネームを作成します。ニックネームとは、アクセス先のデータ・ソース側にあるオブジェクトを参照するために使用される ID です。ニックネームが示すオブジェクトを、データ・ソース・オブジェクトといいます。

別名が代替名であるのとは異なり、ニックネームはデータ・ソース・オブジェクトの代替名ではありません。ニックネームは、フェデレーテッド・サーバーがこれらのオブジェクトを参照するために使用するポインターです。ニックネームは通常、CREATE NICKNAME ステートメントに、特定のニックネーム列オプションとニックネーム・オプションを指定して定義されます。

エンド・ユーザーまたはクライアント・アプリケーションが分散要求をフェデレーテッド・サーバーにサブミットする場合、その要求にデータ・ソースを指定する必要はありません。その代わりに、要求はデータ・ソース・オブジェクトをそのオブジェクトのニックネームで参照します。ニックネームはデータ・ソースの特定のオブジェクトにマップされます。このようにマッピング (対応付け) されることにより、ニックネームをデータ・ソース名で修飾する必要がなくなります。エンド・ユーザーまたはクライアント・アプリケーションは、データ・ソース・オブジェクトのロケーションを意識する必要がありません。

ここで、ニックネーム *DEPT* が、*NFX1.PERSON* という Informix® データベース表を表すように定義するとします。SELECT * FROM *DEPT* というステートメントをフェデレーテッド・サーバーから使用できます。しかし、フェデレーテッド・サーバーに *NFX1.PERSON* というローカル表がなければ、フェデレーテッド・サーバーから SELECT * FROM *NFX1.PERSON* というステートメントを使用することはできません (パススルー・セッションは除く)。

データ・ソース・オブジェクトにニックネームを作成すると、オブジェクトについてのメタデータがグローバル・カタログに追加されます。照会オプティマイザーは、このメタデータとラッパー内の情報を使用して、データ・ソース・オブジェクトへのアクセスを容易にします。たとえば、索引を持つ表にニックネームを作成すると、グローバル・カタログにはその索引についての情報が入ります。ラッパーには、DB2® のデータ・タイプとデータ・ソースのデータ・タイプとの間のマッピングが入っています。

現時点では、一部の DB2 UDB ユーティリティー操作はニックネームに対して実行することができません。

Cross Loader ユーティリティーを使用してニックネームに対してクロス・ロードを実行することはできません。

関連概念:

- 19 ページの『ニックネーム列オプション』

関連タスク:

- 71 ページの『データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』
- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』
- 18 ページの『有効なデータ・ソース・オブジェクト』

有効なデータ・ソース・オブジェクト

アクセスするデータ・ソースのオブジェクトは、ニックネームで識別します。次の表は、フェデレーテッド・システム内のニックネームを作成できるオブジェクトのタイプを示しています。

表 3. 有効なデータ・ソース・オブジェクト

データ・ソース	有効なオブジェクト
DB2 for Linux, UNIX, and Windows	ニックネーム、マテリアライズ照会表、表、ビュー
DB2 for z/OS and OS/390	表、ビュー
DB2 for iSeries	表、ビュー
DB2 for VM and VSE	表、ビュー
Informix	表、ビュー、シノニム
Microsoft SQL Server	表、ビュー
ODBC	表、ビュー
Oracle	表、ビュー、シノニム
Sybase	表、ビュー
Teradata	表、ビュー
BLAST	BLAST 検索アルゴリズム用に索引が付けられた FASTA ファイル
BioRS	BioRS データ・バンク
Documentum	Documentum Docbase 内のオブジェクトおよび登録された表
Entrez	Entrez データベース
Extended Search	データ・ソース (Lotus Notes データベース、Microsoft Access、Microsoft Index Server、Web 検索エンジン、および LDAP ディレクトリーなど) からのファイル

表3. 有効なデータ・ソース・オブジェクト (続き)

データ・ソース	有効なオブジェクト
HMMER	HMMER の hmmpfam または hmmsearch プログラムで検索可能な HMM データベース・ファイル (PFAM などの Hierarchical Markov Models のライブラリー)。
Microsoft Excel	.xls ファイル (ワークブック内の最初のシートだけがアクセスされる)
表構造ファイル	特定のフォーマットに適合するテキスト・ファイル
Websphere Business Integration アダプター	SAP の BAPI、Siebel のビジネス・コンポーネント、および PeopleSoft のコンポーネント・インターフェースにマップされる Websphere Business Integration ビジネス・オブジェクト
Web サービス	Web サービス記述言語ファイル内の操作
XML タグ・ファイル	XML 文書内の項目のセット

関連概念:

- 17 ページの『ニックネームとデータ・ソース・オブジェクト』
- 19 ページの『ニックネーム列オプション』

ニックネーム列オプション

グローバル・カタログには、ニックネームが付けられたオブジェクトに関する追加のメタデータ情報を入れることができます。このメタデータは、データ・ソース・オブジェクトの特定の列の値を記述したものです。このメタデータを、ニックネーム列オプション というパラメーターに割り当てます。ニックネーム列オプションは、列内のデータを通常の列とは異なる方法で処理するようラッパーに指示します。SQL コンパイラーと照会オプティマイザーは、メタデータを使用して、データにアクセスするためのよりよいプランを作成します。

ニックネーム列オプションは、ラッパーにその他の情報を提供するためにも使用されます。たとえば XML データ・ソースの場合、ニックネーム列オプションは、ラッパーが XML 文書から列を解析するとき使用する XPath 式をラッパーに指示するために使用されます。

フェデレーションを使用すると、DB2[®] サーバーはニックネームが参照するデータ・ソース・オブジェクトを、あたかもローカル DB2 表であるかのように扱います。したがって、ニックネームを作成するどのデータ・ソース・オブジェクトに対しても、ニックネーム列オプションをセットすることができます。ニックネーム列オプションの中には特定のタイプのデータ・ソース用に作られたものもあり、それらは該当するデータ・ソースにのみ適用できます。

フェデレーテッド・データベースの照合シーケンスとは異なる照合シーケンスを持つデータ・ソースがあるとします。フェデレーテッド・サーバーは通常、文字データを含む列をデータ・ソース側でソートすることはありません。データはフェデレ

ード・データベースに戻され、ローカルにソートが行われます。しかしここで、列が文字データ・タイプ (CHAR または VARCHAR) であり、数字 ('0'、'1'、...、'9') だけが入っているとします。このことを表すには、NUMERIC_STRING ニックネーム列オプションを 'Y' にします。これにより、DB2 照会オプティマイザーは、オプションでデータ・ソース側でソートを実行できるようになります。ソートをリモート側で実行できれば、データをフェデレーテッド・サーバーに持ってきて、ソートをローカルで実行するというオーバーヘッドが避けられます。

ALTER NICKNAME ステートメントを使用することにより、リレーショナル・ニックネームにニックネーム列オプションを定義することもできます。非リレーショナル・ニックネームには、CREATE NICKNAME および ALTER NICKNAME ステートメントを使用してニックネーム列オプションを定義できます。

関連概念:

- 20 ページの『データ・タイプ・マッピング』

関連タスク:

- 「フェデレーテッド・システム・ガイド」の『ニックネームを使用した操作』

関連資料:

- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』

データ・タイプ・マッピング

フェデレーテッド・サーバーがデータ・ソースからデータを検索するには、データ・ソース側のデータ・タイプが、対応する DB2® のデータ・タイプに対応付けられて (マッピングされて) いなければなりません。デフォルトのデータ・タイプ・マッピングの例として、以下のものがあります。

- Oracle タイプ FLOAT は DB2 タイプ DOUBLE にマップされます
- Oracle タイプ DATE は DB2 タイプ TIMESTAMP にマップされます
- DB2 for z/OS™ タイプ DATE は DB2 タイプ DATE にマップされます

ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 データ・ソース用のデフォルトのタイプ・マッピングは、DRDA® ラッパーにあります。Informix® 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーにあります。その他のタイプ・マッピングについても同様です。

非リレーショナルのデータ・ソースの中には、CREATE NICKNAME ステートメントでデータ・タイプ情報を指定しなければならないものがあります。ニックネームの作成時に、データ・ソース・オブジェクトの列ごとに、対応する DB2 for Linux, UNIX®, and Windows® データ・タイプを指定する必要があります。それぞれの列は、データ・ソース・オブジェクト内の特定のフィールドまたは列にマップされている必要があります。

リレーショナル・データ・ソースの場合は、デフォルトのデータ・タイプ・マッピングをオーバーライドできます。たとえば、Informix INTEGER データ・タイプ

は、デフォルトでは DB2 INTEGER データ・タイプにマップします。デフォルト・マッピングをオーバーライドして、Informix の INTEGER データ・タイプを DB2 DECIMAL(10,0) データ・タイプにマップされるようにすることができます。

新しいタイプ・マッピングの作成またはデフォルトのタイプ・マッピングの変更を行ってから、ニックネームを作成する必要があります。そうしないと、タイプ・マッピングの変更前に作成されたニックネームは、新しいマッピングを反映しなくなります。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・サーバーでのデータ・タイプ・マッピング』

関数マッピング

フェデレーテッド・サーバーがデータ・ソース関数を認識するには、その関数が DB2[®] for Linux, UNIX[®], and Windows[®] にある対応する既存の関数にマップされている必要があります。DB2 Information Integrator は、既存の組み込みデータ・ソース関数と DB2 側の対応する組み込み関数の間のデフォルトのマッピングを備えています。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトの関数マッピングがあります。DB2 for z/OS[™] and OS/390[®] 関数への関数へのデフォルトの関数マッピングは、DRDA[®] ラッパーにあります。Sybase 関数へのデフォルトの関数マッピングは CTLIB ラッパーにあります。その他のタイプ・マッピングについても同様です。

リレーショナル・データ・ソースの場合、フェデレーテッド・サーバーが認識しないデータ・ソース関数を使用するには、関数マッピングを作成する必要があります。データ・ソース関数とフェデレーテッド・データベースにある対応する DB2 関数との間に、マッピングを作成します。関数マッピングは通常、新しい組み込み関数や新しいユーザー定義関数がデータ・ソース側で使用可能になったときに使用されます。関数マッピングは、DB2 側に対応する関数がない場合にも使用されます。その場合には、関数テンプレートも作成しなければなりません。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムでの関数マッピング』
- 21 ページの『索引の指定』

索引の指定

データ・ソース表にニックネームを作成すると、データ・ソース表が持つすべての索引についての情報がグローバル・カタログに追加されます。照会オプティマイザーはこの情報を使用して、分散要求の処理を効率よく行います。データ・ソース索引についてのカタログ情報はメタデータの集まりであり、索引の指定 と呼ばれます。フェデレーテッド・サーバーは、次のものにニックネームが作成された場合、「索引の指定」を作成しません。

- 索引を持たない表

- ビュー。通常、ビューにはリモート・カタログに保管される索引情報はありません
- フェデレーテッド・サーバーが索引情報を入手できるリモート・カタログを持たない、データ・ソース・オブジェクト

ニックネームの作成時にあった索引に加えて、新しい索引が表に追加されたとします。索引情報はニックネーム作成時にグローバル・カタログに入るため、フェデレーテッド・サーバーは新しい索引については認識していません。同様に、ビューのニックネームを作成しても、フェデレーテッド・サーバーはそのビューの基になる表（およびその索引）を認識していません。このような場合には、ユーザーが必要な索引情報をグローバル・カタログに入れることができます。索引を持たない表の「索引の指定」を作成することができます。「索引の指定」は、データを速く見つけるには表内のどの列を検索すべきかを照会オプティマイザーに示します。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムでの索引の指定』

フェデレーテッド・システムと対話する方法

フェデレーテッド・データベースは DB2[®] Universal Database であるため、フェデレーテッド・システムと対話するには、以下のいずれかの方法を使用できます。

- DB2 コマンド行プロセッサ (CLP)
- DB2 コマンド・センター GUI
- DB2 コントロール・センター GUI
- アプリケーション・プログラム
- DB2 ファミリー・ツール
- Web サービス・プロバイダー

フェデレーテッド・データベースの資料で説明されている手順には、DB2 コマンド行プロセッサまたは DB2 コマンド・センター GUI に入力するコマンドおよび SQL ステートメントが示されています。資料が説明しているのは、DB2 コントロール・センター GUI を使用して作業を行うタイミングです。しかし、DB2 コントロール・センター GUI の操作は見れば分かるものなので、本書には DB2 コントロール・センターを使用してこれらの作業を実行するための手順は含まれていません。

DB2 コマンド行プロセッサ (CLP)

フェデレーテッド・システムのセットアップ、構成、調整、および保守に必要なすべての作業は、DB2 コマンド行プロセッサから実行することができます。場合によっては、DB2 コマンド行プロセッサまたは DB2 コマンド・センターを使用しなければならないことがあります。たとえば、次のような場合です。

- ユーザー定義データ・タイプ・マッピングを作成、変更、またはドロップする
- ユーザー定義関数マッピングを作成、変更、またはドロップする

DB2 コマンド・センター

DB2 コマンド・センターを使用すれば、長い SQL ステートメントを手作業で入力しなくても、分散要求を作成および実行することができます。フェデレーテッド・システムのパフォーマンスをチューニングする場合は、DB2 コマンド・センターを使用します。DB2 コマンド・センターは、分散要求のアクセス・プランを見るために DB2 Explain の機能を使用する場合に便利です。DB2 コマンド・センターは、SQL アシスタント・ツールで作業する場合にも使用できます。

DB2 コントロール・センター

DB2 コントロール・センター GUI を使用すると、フェデレーテッド・システムのセットアップ、構成、および変更に必要な作業のほとんどを実行することができます。DB2 コントロール・センターでは、パネル (ダイアログ・ボックスとウィザード) を使って作業を進めていきます。これらのパネルには、リスト・ボックスやコマンド・ボタンなどをマウスでポイントすると表示される、対話式ヘルプが組み込まれています。さらに各パネルには、パネルでの作業についての情報を提供する「ヘルプ」ボタンがあり、関係する概念や参照情報にリンクしています。

フェデレーテッド・オブジェクトは、ウィザードを使用して作成するか、または各オブジェクトを個別に作成することができます。

Web サービス、WebSphere® Business Integration、および XML データ・ソースへのアクセスを構成するには、DB2 コントロール・センターを使用します。DB2 コントロール・センターに組み込まれた機能を使用すると、簡単な手順でフェデレーテッド・サーバーを構成して、これらのデータ・ソースにアクセスさせることができます。

DB2 コントロール・センター GUI は、次のような基本的なデータ・ソース構成作業を行う最も簡単な方法です。

- ラッパーを作成し、ラッパー・オプションを設定する
- データ・ソースの環境変数を指定する
- サーバー定義を作成し、サーバー・オプションを設定する
- ユーザー・マッピングを作成し、ユーザー・オプションを設定する
- ニックネームを作成し、ニックネーム・オプションまたは列オプションを設定する

データ・ソースにアクセスできるようにフェデレーテッド・サーバーを構成した後、DB2 コントロール・センターを使用して次のことを行えます。

- データ・ソース構成の変更
- ニックネームとサーバーの状況のモニター
- ニックネームの現在の統計の保守
- キャッシュ表の作成および変更
- ニックネームの情報制約の指定
- 透過 DDL を使用した DB2 Information Integrator によるリモート表の作成

アプリケーション・プログラム

アプリケーションに、フェデレーテッド・データを使用するための特別なコーディングは必要ありません。アプリケーションは、他の DB2 クライアント・アプリケーションとまったく同様にシステムにアクセスします。アプリケーションは、フェデレーテッド・サーバー内にあるフェデレーテッド・データベースとインターフェースをとります。データ・ソースからデータを入手するために、アプリケーションは DB2 SQL で書かれた照会をフェデレーテッド・データベースにサブミットします。すると DB2 Information Integrator は、その照会を該当するデータ・ソースに配布し、要求されたデータを収集し、そのデータをアプリケーションに戻します。ただし、DB2 Information Integrator はニックネームを使用してデータ・ソースとやりとりするので、ユーザーは次のことに注意する必要があります。

- ニックネームを使用する場合の SQL の制約事項
- ニックネームが付けられたオブジェクトの操作方法

DB2 ファミリー・ツール

フェデレーテッド・データベースと対話するために、以下のようなホストおよびミッドレンジ・ツールを使用することもできます。

- DB2 SPUFI (DB2 for z/OS™ and OS/390® の場合)
- 対話式 SQL (STRSQL) (DB2 for iSeries™ の場合)

Web サービス・プロバイダー

Web サービス・ラッパーを使用し、Web サービス・プロバイダー経由でフェデレーテッド・データベースと対話することもできます。

関連概念:

- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』

第 2 章 フェデレーテッド・データ・ソースの構成の計画

以下の各セクションでは、フェデレーテッド・システムの計画作業に役立つ情報を示します。

フェデレーテッド・オブジェクトの命名規則

他の DB2 オブジェクトと同様に、フェデレーテッド・データベース・オブジェクトにも命名規則があります。

フェデレーテッド・データベース・オブジェクトには、以下のものが含まれます。

- 関数マッピング
- 索引の指定
- ニックネーム
- サーバー
- タイプ・マッピング
- ユーザー・マッピング
- ラッパー

フェデレーテッド・オブジェクト名は、下記のいずれかで始まらなければなりません。

- 文字、有効なアクセント付き文字も含まれます。(例: Ö)
- マルチバイト文字、マルチバイト・スペースは含みません。(マルチバイト環境用)

フェデレーテッド・オブジェクト名は、数字または下線文字で始めることはできません。

フェデレーテッド・オブジェクト名には、次の文字も含めることができます。

- A ~ Z
- 0 ~ 9
- @、#、\$、および _ (下線)

フェデレーテッド・オブジェクト名は 128 バイトを超えることはできません。

オプション (サーバー・オプションやニックネーム・オプションなど) やオプション設定値は、255 バイトまでです。

引用符の付かない名前は、大文字に変換されます。

関連概念:

- 「管理ガイド: インプリメンテーション」の『NLS 環境での命名規則』
- 「管理ガイド: インプリメンテーション」の『Unicode 環境での命名規則』

関連資料:

- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』
- 26 ページの『フェデレーテッド・システムでの大文字小文字の区別をする値の保持』

フェデレーテッド・システムでの大文字小文字の区別をする値の保持

フェデレーテッド・システムで、データ・ソースで大文字小文字が区別されるユーザー ID やパスワードなど値を指定する必要が生ずることがよくあります。これらの値がデータ・ソースに渡された際に大文字小文字が正しく伝えられるためには、以下のガイドラインに沿って行ってください。

- その値の大文字小文字を正しく指定して、正しい引用符で囲む。二重引用符は、ラッパーやニックネームなどのオブジェクト名の場合はオプションです。REMOTE_AUTHID や REMOTE_PASSWORD ユーザー・マッピング・オプションなどのオプションの値には、単一引用符が必要です。
- ユーザー ID とパスワードの場合、FOLD_ID と FOLD_PW サーバー・オプションを指定してこれらの値を自動的に正しい大文字小文字に変換させることができます。このオプションを使えば、各データ・ソースで必要な大文字小文字を覚えている必要はありません。大文字小文字を気にせずに入力しても自動的に変換されます。

UNIX オペレーティング・システムのコマンド・プロンプトからの場合:

フェデレーテッド・サーバーのオペレーティング・システムのコマンド・プロンプトで大文字小文字の区別をする値を引用符で囲む場合、引用符が正しく構文解析されるようにする必要があります。

- SQL ステートメントに二重引用符が含まれるが、単一引用符が含まれない場合、ステートメント全体を単一引用符で囲みます。たとえば、次の SQL ステートメントを発行するとします。

```
CREATE NICKNAME my_nick FOR my_server."owner"."my_table"
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
DB2 'CREATE NICKNAME my_nick FOR my_server."owner"."my_table"'
```

- SQL ステートメントに単一引用符が含まれるが、二重引用符が含まれない場合、ステートメント全体を二重引用符で囲みます。たとえば、次の SQL ステートメントを発行するとします。

```
CREATE USER MAPPING FOR USER SERVER my_server
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
DB2 "CREATE USER MAPPING FOR USER SERVER my_server
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')"
```

- SQL ステートメントに単一引用符と二重引用符の両方が含まれる場合は、以下のようにします。
 - ステートメント全体を二重引用符で囲みます。
 - 二重引用符が必要な値の前に円記号を付けます。

たとえば、次の SQL ステートメントを発行するとします。

```
CREATE USER MAPPING FOR "local_id" SERVER my_server
  OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

UNIX コマンド・プロンプトで次のようにテキストを入力します。

```
DB2 "CREATE USER MAPPING FOR ¥"local_id¥" SERVER my_server
    OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')"
```

上述の例では、SQL ステートメントを UNIX コマンド・プロンプトから入力し、ステートメントを DB2 コマンドへ `-f` オプションなしで渡していると仮定しています。SQL ステートメントを `f-` オプション付きの DB2 コマンドを使用してファイルから入力する場合は、それぞれの例の最初のかたちでステートメントを入力しなければなりません。

Windows オペレーティング・システムのコマンド・プロンプトからの場合:

Windows で大文字小文字を区別する値を保持するには、各引用符の前に円記号を置きます。たとえば、`weekly_salary`という Microsoft SQL サーバーの表用に `NICK1` というニックネームを作りたいとします。この表は、`NORBASE` データベースにあります。ローカル・スキーマは、`my_schema` です。

フェデレーテッド・サーバーの Windows コマンド・プロンプトで次のように入力します。

```
DB2 CREATE NICKNAME nick1
    FOR norbase.¥"my_schema¥".¥"weekly_salary¥"
```

DB2 CLP またはアプリケーション・プログラムからの場合:

DB2 コマンド行プロンプト (CLP) から、またはアプリケーション・プログラムで値を指定する場合、正しい引用符で囲むことによって大文字小文字を区別する値を保持することができます。

たとえば、ユーザー ID `local_id` のためのユーザー・マッピングを作成したいとします。リモート・ユーザー ID が `my_id` で、リモート・パスワードが `my_password` です。これら 3 つの値をみな小文字のまま保持したいとします。DB2 コマンド・プロンプトで次のように入力します。

```
| CREATE USER MAPPING FOR "local_id" SERVER my_server
|     OPTIONS(REMOTE_AUTHID 'my_id', REMOTE_PASSWORD 'my_password')
```

関連資料:

- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 647 ページの『付録 E. フェデレーテッド・システムのユーザー・マッピング・オプション』
- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』

データ・ソース統計の更新

| リレーショナル・データ・ソースにアクセスする計画なら、そのデータ・ソースに
| アクセスするフェデレーテッド・サーバーを構成する前に、リモート・データ・ソ
| ースで統計を更新しなければなりません。リモート・データ・ソースの統計を最新
| にしておくと、照会のパフォーマンスを向上させることができます。

フェデレーテッド・サーバーは、照会処理を最適化するのに、フェデレーテッド・データベースに保存されているデータ・ソース統計に依存します。これらの統計

は、データ・ソース・オブジェクトにニックネームを作成した時点で収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するか検査し、次にそのデータ・ソースの既存の統計データを収集しようとします。照会オプティマイザーにとって有用な情報は、データ・ソース・カタログから読み込まれ、フェデレーテッド・データベースのシステム・カタログに追加されます。照会オプティマイザーはデータ・ソースからのカタログ情報の一部またはすべてを使用することがあるため、ニックネームを作成する前に、データ・ソース側で統計を更新することをお勧めします。DB2 RUNSTATS コマンドに相当するコマンドをデータ・ソース側で使用して、データ・ソースの統計を更新します。

オブジェクトのニックネームを作成する際に、フェデレーテッド・データベースはデータ・ソース・オブジェクトのリモート統計情報を検索します。ニックネームの作成後に、データ・ソースがオブジェクトのカタログ統計を更新した場合、統計情報の変更はフェデレーテッド・データベースのシステム・カタログには伝搬されません。フェデレーテッド・データベース内のシステム・カタログがリモート・データ・ソース・オブジェクトの現在の統計を反映するようにするには、フェデレーテッド・サーバーが統計を更新するように要求しなければなりません。

アクション: アクセスしたいデータ・ソース・オブジェクトを識別します。これらはニックネームを作成する対象のオブジェクトです。これらのオブジェクトがその部分であるデータ・ソースのうち、どれが統計を更新できるかを判断します。プランニング・チェックリストのデータ・ソース統計表にそれらのデータ・ソースをリストします。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『ニックネーム統計の更新機能 - 概要』

関連資料:

- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』

正しいラッパーの選択

ほとんどのデータ・ソースの場合、そのデータ・ソースにアクセスするのに使用できるラッパーは 1 つしかありません。しかしながら、データ・ソース内のデータにアクセスするのに、どのラッパーを使用するか選択することができるデータ・ソースもあります。

ODBC API をサポートするデータ・ソースにアクセスするには、これらのデータ・ソース用に設計されたラッパーを使用することも、ODBC ラッパーを使用することもできます。こうしたデータ・ソースの例には、Oracle、Microsoft Excel、Microsoft SQL Server などがあります。通常は、これらのデータ・ソース専用設計されたラッパーを使用したほうが、照会のパフォーマンスは向上します。

ODBC ドライバーはあるものの、DB2 Information Integrator に組み込まれている特定のデータ・ソース・ラッパーではサポートされていないデータ・ソースへのアクセスには、ODBC ラッパーを使用します。たとえば、RedBrick データ・ソースにアクセスするには、ODBC ラッパーを使用します。

DB2 for Linux、UNIX、および Windows のデータ・ソース

DB2 Universal Database for Linux, UNIX, Windows データ・ソースにアクセスするために、ODBC ラッパーは使用しないでください。 DB2 Universal Database for Linux, UNIX, Windows データ・ソースにアクセスするために、ODBC ラッパーは使用することはサポートされていません。 DB2 Universal Database for Linux, UNIX, Windows データ・ソースにアクセスするには、DRDA ラッパーを使用してください。

Excel データ・ソース

必要に応じて、Excel ラッパーを使用する代わりに ODBC ラッパーを使用して Excel データにアクセスできます。

Informix データ・ソース

Informix データ・ソースにアクセスするには、ODBC ラッパーは使用しないでください。 ODBC ラッパーを使用して Informix データ・ソースにアクセスすることはサポートされていません。 Informix データ・ソースにアクセスするには、Informix ラッパーを使用します。

アクション: プランニング・チェックリストのラッパー表にフェデレーテッド・システム用に作成するラッパーを記入します。

関連概念:

- 29 ページの『Excel データにアクセスする方法』

関連タスク:

- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』
- 341 ページの『フェデレーテッド・サーバーへの ODBC データ・ソースの追加』

関連資料:

- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』

Excel データにアクセスする方法

Excel ラッパーまたは ODBC ラッパーのどちらかを使用して、Microsoft® Excel ワークシートのデータにアクセスできます。

Excel データを照会するには、両方のラッパーとも、Excel ワークブック中のワークシートを開いて読み取れる DB2® フェデレーテッド・サーバーが必要です。それで、Excel ワークブックは、フェデレーテッド・サーバーと同じコンピューター上か、またはネットワーク・アクセス可能なドライブ上になければなりません。

Excel ラッパーを使用する場合は、Excel アプリケーションがフェデレーテッド・サーバー上にインストールされていなければなりません。

ODBC ラッパーを使用する場合は、Excel ODBC ドライバーがフェデレーテッド・サーバー上になければなりません。このドライバーは、自動的に Microsoft Windows® と共にインストールされます。Excel アプリケーションをフェデレーテッド・サーバー上にインストールする必要はありません。

各ラッパーには、Excel ワークブック中のデータの種類とレイアウトに関するいくつかの要件があります。Excel ラッパーを使用する場合は、ワークブック中の最初のワークシート中のデータしかアクセスできません。ODBC ラッパーを使用する場合は、ワークブック中のどのワークシートのデータでもアクセスできます。

以下の例は、これらの 2 つのラッパーに関するワークシートのレイアウトの要件を示しています。

複数のラベル行と数式があるワークシートの例:

この例は、ワークシートの上部に複数のラベル行があり、ブランク行を含み、13 行目に数式がある例を示しています。このワークシートのデータにアクセスするには、アクセスしたいセルの範囲を示さなければなりません。

	A	B	C	D
1	コンパウンドの分析			
2				
3	コンパウンド名	重み	分子のカウント	テスト済みかどうか
4	compound_A	1.23	367	テスト済み
5	compound_G		210	
6	compound_F	0.000425536	174	テスト済み
7	compound_Y	1.000256		テスト済み
8	compound_Q		1024	
9	compound_B	33.5362		
10	compound_S	0.96723	67	テスト済み
11	compound_O	1.2		テスト済み
12				
13	テスト済みのコンパウンドの合計			5

図 2. 複数のラベル行と数式を含むワークシート

Excel ラッパーを使用する場合

CREATE NICKNAME ステートメント中でセルの範囲を指定するには、RANGE オプションを使用します。指定する範囲には、データだけを含めてください。列ラベルをこの範囲に含めないでください。SUM などの数式を含むセルは、数式ではなく数式の結果を戻します。数式の結果を戻したいのでなければ、数式を含むセルをこの範囲に含めないでください。この例では、セルの範囲として A4:D11 を RANGE オプションに含めます。

ODBC ラッパーを使用する場合

セルの範囲に名前を作成して、ワークシート中のデータの種類を明示的に指定しなければなりません。Excel は、このセルの範囲を名前付き範囲として参照します。Excel ODBC ドライバーは、範囲内の先頭行である 1 行の

ラベル行のみ認識します。ラベルとデータの間には空白行は許されません。名前付き範囲には、列ラベルの行を 1 行だけ含めなければなりません。CREATE NICKNAME ステートメントで名前付き範囲を指定します。名前を付ける範囲中には、列ラベルの行は 1 行だけ含めなければなりません。名前付き範囲内に 1 つの列ラベル行を含めない場合は、最初のデータ行が列ラベルとして扱われます。SUM などの数式を含むセルは、数式ではなく数式の結果を返します。数式の結果を戻したいのであれば、数式を含むセルをこの範囲に含めないでください。この例では、名前を付けるセルの範囲は A3:D11 です。

1 つのラベル行を含むワークシートの例:

この例は、ワークシートの上部に 1 つの列ラベル行のみある例を示しています。このレイアウトには、追加のラベル行、空白行、数式を含むセルはありません。

図 3. 1 行目に 1 つの列ラベル行を含むワークシート

Excel ラッパーを使用する場合

CREATE NICKNAME ステートメント中でセルの範囲を指定するには、RANGE オプションを使用しなければなりません。この範囲に 1 行目の列ラベルを含めることはできません。セルの範囲として A2:D9 を指定します。

ODBC ラッパーを使用する場合

名前付き範囲を作成せずにこのデータにアクセスできます。CREATE NICKNAME ステートメント中にワークシート名を指定します。ラッパーは最初の非空白行をラベルとして読み取り、その情報をニックネームのための列名として使用します。後続の行はデータとして読み取られます。

データのみを含むワークシートの例:

この例は、データのみを含むワークシートを示しています。列ラベルの行、ブランク行、公式を含むセルはありません。

	compound_A	1.23	367	テスト済み
	compound_G		210	
	compound_F	0.000425536	174	テスト済み
	compound_Y	1.000256		テスト済み
	compound_Q		1024	
	compound_B	33.5362		
	compound_S	0.96723	67	テスト済み
	compound_O	1.2		テスト済み

図4. データのみを含むワークシート

Excel ラッパーを使用する場合

ワークブック中の最初のワークシート中にデータがある場合は、RANGE オプションを使用しなくてもこのラッパーはデータにアクセスできます。ワークブック中のそれ以外のワークシート中にデータがある場合は、CREATE NICKNAME ステートメント中に RANGE オプションを指定しなければなりません。

ODBC ラッパーを使用する場合

ODBC ラッパーを使用して Excel データにアクセスする場合は、このラッパーは Excel ODBC ドライバーのサポート内容による制限を受けます。Excel ODBC ドライバーには、特定のワークシートの形式が必要になります。このドライバーは、最初の非ブランク行に列ラベルが含まれていることを想定しています。最初の非ブランク行にデータが含まれている場合は、この行のデータは残りのデータの列ラベルとして扱われます。ワークシートに列ラベルの行が含まれていない場合、最初の行はデータではなくラベルとして使用されます。その結果、最初のデータ行は失われます。ワークシートに変更を加えると、この要件を満たすことができます。データの前に新しい行を挿入し、各データ列のラベルを追加すると、1つのラベル行を含む例のようになります。

関連タスク:

- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』
- 353 ページの『ODBC ラッパーを使用した Excel データへのアクセス』

ユーザー・マッピングの計画

フェデレーテッド・サーバーがデータ・ソースに要求をプッシュダウンする必要がある場合、サーバーはまずデータ・ソースに接続を確立する必要があります。データ・ソースによっては、フェデレーテッド・サーバーはそのデータ・ソースに対して有効なユーザー ID とパスワードを使用して、接続を確立します。こうしたデータ・ソースの場合、フェデレーテッド・サーバーのユーザー ID とパスワードおよびデータ・ソースのユーザー ID とパスワードの間の関連付けを定義しておく必要があります。分散要求を送信するためにフェデレーテッド・システムを使用するユーザー ID ごとに、この関連付けを必ず作成しておかなければなりません。この関連付けをユーザー・マッピング といいます。

1 つのデータ・ソースに同じユーザー ID とパスワードでアクセスするユーザーのグループに対して、DB2 コントロール・センターを使用してユーザー・マッピングを作成することができます。

アクション: フェデレーテッド・サーバーとデータ・ソースとの間のユーザー・マッピングの作成が必要なユーザー ID を洗い出します。プランニング・チェックリストのユーザー・マッピング表に、フェデレーテッド・サーバーのユーザー ID と対応するデータ・ソースのユーザー ID をリストします。

関連資料:

- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』

データ・タイプ・マッピングの計画

データ・ソースのデータ・タイプは、リモート・データ・タイプと呼ばれ、フェデレーテッド・データベースのデータ・タイプは、ローカル・データ・タイプと呼ばれます。

データ・ソースによっては、データ・ソースのデータ・タイプとフェデレーテッド・データベースの間のデフォルト・マッピングがラッパーにあります。データ・ソース・オブジェクトに対するニックネームの作成時に、列に関する情報はフェデレーテッド・データベースのシステム・カタログに保存されます。列のデータ・タイプは、デフォルトの順方向データ・タイプ・マッピングに基づきます。

他のデータ・ソースでは、ニックネーム作成時に列情報とデータ・タイプを指定しなければなりません。ある種の非リレーショナル・ラッパーは、データ・ソースにアクセスするために必要なすべての列を作成します。これらは**固定列**と呼ばれます。他の非リレーショナル・データ・ソースでは、列のデータ・タイプの一部またはすべてを指定することができます。

アプリケーションによっては、デフォルトのマッピングとは異なるデータ・タイプ・マッピングを必要とする場合があります。データ・タイプ・マッピングが指定できるラッパーの場合、デフォルト・マッピングをオーバーライドして以下を行うことができます。

- 特定のサーバーにあるすべてのデータ・ソース・オブジェクトについて、タイプ・マッピングの変更

- 特定のデータ・ソース・オブジェクトのタイプ・マッピングの変更
- 特定のデータ・ソース・タイプのタイプ・マッピングの変更
- 特定のデータ・ソース・タイプおよびバージョンのためのタイプ・マッピングの変更

新しいデータ・タイプ・マッピングを定義するには、CREATE TYPE MAPPING ステートメントを使用します。作成したマッピングは、フェデレーテッド・データベースのグローバル・カタログ SYSCAT.TYPEMAPPINGS ビューに保管されます。

データ・タイプ・マッピングの変更は、データ・ソース・オブジェクトのニックネームを作成する前に行います。データ・ソース・オブジェクトにニックネームを作成すると、フェデレーテッド・サーバーは表についての情報をグローバル・カタログに入れます。この情報には、ニックネーム、データ・ソースの表名、表のそれぞれの列に定義された列名およびデータ・タイプが含まれます。

マッピングが変更された後に作成されたニックネームのみが、新しいタイプ・マッピングを反映します。マッピングを変更する前に作成されたニックネームは、デフォルトのデータ・タイプ・マッピングを使用します。

ニックネームを作成した後でデータ・タイプ・マッピングを作成した場合は、新しいマッピングを反映するように各ニックネームを変更するか、ニックネームをドロップして再作成する必要があります。

注: データ・ソース表が特殊データ・タイプまたはユーザー定義データ・タイプの列を含む場合、次の 2 つの選択があります。

- フェデレーテッド・データベースにタイプ・マッピングを作成した後、そのデータ・ソース表のニックネームを作成することができます。ニックネームを作成する前にタイプ・マッピングを作成することにより、フェデレーテッド・サーバーはどのデータ・タイプをこれらの列にマップするべきかを知ることができます。これらの特殊データ・タイプまたはユーザー定義データ・タイプのマッピングを CREATE NICKNAME ステートメントを出す前に作成しておかないなら、エラーになります。
- データ・ソース表の列が以下の条件のいずれかを満たす場合、
 - 列は、システムまたは組み込みのデータ・タイプに基づくユーザー定義データ・タイプである
 - 列は、データ・タイプ・マッピングでサポートされない属性を持つ

列を基になる組み込みデータ・タイプに関連付けるかキャストするビューをデータ・ソースに作成することができます。次に、その表ではなくビューに対するニックネームを作成します。

アクション: 新しいマッピングを定義したいデータ・タイプ・マッピングを決定します。作成したいデータ・ソースとタイプ・マッピングを、プランニング・チェックリストのデータ・タイプ・マッピング表にリストします。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・サーバーでのデータ・タイプ・マッピング』

関連資料:

- 35 ページの『フェデレーテッド・システム構成の計画のためのチェックリスト』
- 569 ページの『非リレーショナル・データ・ソースでサポートされるデータ・タイプ』

関数マッピングの計画

DB2 for UNIX、および DB2 for Windows は、既存の組み込みデータ・ソース関数と組み込み DB2 関数間のデフォルトの関数マッピングを備えています。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトの関数マッピングがあります。非リレーショナルのデータ・ソースの中には、デフォルトの関数マッピングを変更できないものもあります。

フェデレーテッド・サーバーが認識しないデータ・ソース関数を使用するには、関数マッピングを作成する必要があります。作成するマッピングは、データ・ソース関数と、フェデレーテッド・データベースの対応する関数との間のものです。関数マッピングは通常、新しい組み込み関数または新しいユーザー定義関数がデータ・ソース側で使用可能になった時に使用されます。

関数マッピングは、DB2 側に対応する関数がない場合にも使用されます。この場合、関数マッピングを作成する前に、フェデレーテッド・データベースに関数テンプレートを作成する必要があります。

アクション: データ・ソースに関数マッピングを作成する必要があるかどうかを判断します。必要な関数マッピングをプランニング・チェックリストの関数マッピング表にリストします。

関連概念:

- 21 ページの『関数マッピング』

フェデレーテッド・システム構成の計画のためのチェックリスト

このプランニング・チェックリストに従って進めることにより、フェデレーテッド・システムの構成が容易になります。このチェックリストは、フェデレーテッド・システムの構成を最適化する方法のガイドとなるものです。

チェックリスト: フェデレーテッド・オブジェクトのネーミング規則

フェデレーテッド・オブジェクトのネーミング規則を知っていますか。

フェデレーテッド・オブジェクトのネーミング規則についての情報は、このトピックの終わりの関連リンクを見て探してください。

チェックリスト: 大文字小文字の区別をする値の保持

データ・ソースに送信するユーザー ID とパスワードの値の大文字小文字を保持するために FOLD_ID および FOLD_PW サーバー・オプションを設定しますか。次の表を使用して、これらのオプションをどのサーバー定義に適用するかを示します。

表4. プランニング・チェックリスト: フェデレーテッド・システムに設定する FOLD_ID および FOLD_PW サーバー・オプション

データ・ソース	このデータ・ソースのサーバー定義で、サーバーにどのような名前を指定しますか?	FOLD_ID サーバー・オプションにどのような設定を指定しますか?	FOLD_PW サーバー・オプションにどのような設定を指定しますか?

チェックリスト: データ・ソース統計

次の表にフェデレーテッド・システムの一部となるデータ・ソースをリストします。データ・ソースへアクセスするためにフェデレーテッド・サーバーを構成する前に、統計を更新するデータ・ソースを示します。下記の表では、例として DB2 UDB for Linux, UNIX, and Windows がリストされています。

表5. プランニング・チェックリスト: フェデレーテッド・システム用に更新するデータ・ソース統計

データ・ソース	このデータ・ソースはカタログ情報を保ちますか? (Y/N)	このデータ・ソースの統計を更新しますか? (Y/N)	統計を更新するためにデータ・ソースが使用するユーティリティーの名前
DB2 for Linux, UNIX, and Windows	Y	Y	RUNSTATS

チェックリスト: データ・タイプ・マッピング

次の表には、マッピングを作成する必要があるデータ・ソースのデータ・タイプと対応するフェデレーテッド・サーバーのデータ・タイプを示します。

表6. プランニング・チェックリスト: フェデレーテッド・システム用に作成するデータ・タイプ・マッピング

データ・ソース	このデータ・ソースのサーバー定義で、サーバーにどのような名前を指定しますか?	データ・ソース・データ・タイプ	フェデレーテッド・サーバーのデータ・タイプ

表6. プランニング・チェックリスト: フェデレーテッド・システム用に作成するデータ・タイプ・マッピング (続き)

データ・ソース	このデータ・ソースのサーバー定義で、サーバーにどのような名前を指定しますか?	データ・ソース・データ・タイプ	フェデレーテッド・サーバーのデータ・タイプ

チェックリスト: ユーザー・マッピング

次の表には、フェデレーテッド・サーバーのユーザー ID とフェデレーテッド・システムの一部となる各 データ・ソース用の対応するユーザー ID を示します。

表7. プランニング・チェックリスト: フェデレーテッド・システム用に作成するユーザー・マッピング

		データ・ソース	データ・ソース	データ・ソース
		_____	_____	_____
ユーザー名	DB2 for Linux, UNIX, and Windows のユーザー ID	ユーザー ID	ユーザー ID	ユーザー ID

チェックリスト: ラッパー

次の表には、作成するラッパーを示します。

表8. プランニング・チェックリスト: フェデレーテッド・システム用に作成するラッパー

データ・ソース	デフォルトのラッパー名	ラッパーに付ける名前
BioRS	なし	
BLAST	なし	
ビジネス・アプリケーション (WebSphere Business Integration ラッパー)	なし	

表 8. プランニング・チェックリスト: フェデレーテッド・システム用に作成するラッパー
(続き)

データ・ソース	デフォルトのラッパー名	ラッパーに付ける名前
DB2 Universal Database™ for Linux, UNIX, and Windows®	DRDA	
DB2 Universal Database for z/OS and OS/390®		
DB2 Universal Database for iSeries		
DB2 Server for VM and VSE		
Documentum	なし	
Entrez	なし	
Excel	なし	
Extended Search	なし	
HMMER	なし	
Informix	INFORMIX	
Microsoft® SQL Server	MSSQLODBC3	
Oracle	NET8	
ODBC	なし	
OLE DB	OLEDB	
Sybase	CTLIB	
表構造ファイル	なし	
Teradata	TERADATA	
Web サービス	なし	
XML	なし	

関連概念:

- 61 ページの『データ・ソースの構成の近道』
- 29 ページの『Excel データにアクセスする方法』

関連タスク:

- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 57 ページの『フェデレーテッド・データベースの作成』

関連資料:

- 51 ページの『フェデレーテッド・データベースの各国語に関する考慮事項』
- 25 ページの『フェデレーテッド・オブジェクトの命名規則』
- 26 ページの『フェデレーテッド・システムでの大文字小文字の区別をする値の保持』
- 27 ページの『データ・ソース統計の更新』
- 33 ページの『データ・タイプ・マッピングの計画』
- 35 ページの『関数マッピングの計画』
- 33 ページの『ユーザー・マッピングの計画』

- 28 ページの『正しいラッパーの選択』

第 2 部 フェデレーテッド・サーバーおよびデータベース

第 3 章 フェデレーテッド・サーバーのセットアップのチェック

フェデレーテッド・サーバーのセットアップのチェック

フェデレーテッド・サーバーのかぎとなる設定をチェックすることにより、構成についての潜在的な問題を避けられます。

手順:

フェデレーテッド・サーバーのセットアップをチェックするには、次のようにします。

- データ・ソース・クライアント・ソフトウェアへのラッパー・ライブラリー・ファイルのリンク・エディットを確認する (UNIX)
- FEDERATED パラメーターが YES に設定されていることを確認する

フェデレーテッド・サーバーのセットアップをチェックした後、フェデレーテッド・データベースを作成する必要があります。

関連タスク:

- 43 ページの『ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)』
- 49 ページの『FEDERATED パラメーターのチェック』
- 57 ページの『フェデレーテッド・データベースの作成』

ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)

ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)

ラッパー・ライブラリー・ファイルのリンク・エディットの確認は、フェデレーテッド・サーバーをセットアップするという、より大きな作業の一部を成しています。

UNIX フェデレーテッド・サーバーでは、データ・ソース用のデータ・ソース・クライアント・ソフトウェアとリンク・エディットする必要のあるラッパーがあります。リンク・エディット・ステップは、DB2 Information Integrator のインストール時に試行されます。リンク・エディット・ステップによって、フェデレーテッド・サーバーが通信するデータ・ソースごとにラッパー・ライブラリーが作成されます。

このタスクは次のデータ・ソースにのみ適用されます。

- Informix
- Microsoft SQL Server
- Oracle

- Sybase
- Teradata

データ・ソースにアクセスするようにフェデレーテッド・サーバーとデータベースを構成する前に、ラッパー・ライブラリー・ファイルのリンク・エディットが正しく行われていることを確認する必要があります。

前提条件:

データ・ソースにアクセスするように適切にセットアップされているフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

手順:

次のようにして、ラッパー・ライブラリー・ファイルのリンク・エディットの状況を判別します。

- リンク・エディットが成功した場合、ラッパー・ライブラリー・ファイルが DB2 Information Integrator がインストールされているディレクトリーに生成されます。
- リンク・エディットが失敗した場合、DB2 Information Integrator がインストールされているディレクトリーにあるエラー・メッセージ・ファイルをチェックしてください。
- リンク・エディットが実行されなかった場合、ライブラリー・ファイルとメッセージ・ファイルのどちらも DB2 Information Integrator がインストールされているディレクトリーに生成されません。リンク・スクリプトを手動で実行する必要があります。

関連タスク:

- 44 ページの『ラッパー・ライブラリー・ファイルのチェック (UNIX)』
- 46 ページの『リンク・エディット・メッセージ・ファイルのチェック (UNIX)』
- 47 ページの『データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする』
- 63 ページの『DB2 UDB コントロール・センターを使用したフェデレーテッド・サーバーへのデータ・ソースの追加』

ラッパー・ライブラリー・ファイルのチェック (UNIX)

ラッパー・ライブラリー・ファイルのチェックは、ラッパー・ライブラリーとデータ・ソース・クライアント・ソフトウェアの間のリンク・エディットを確認するという、より大きな作業の一部を成しています。

ラッパー・ライブラリー・ファイルは、データ・ソースへのアクセスに必要です。データ・ソースの中には、DB2 Information Integrator のインストール時にライブラリー・ファイルがフェデレーテッド・サーバーに追加されるものがあります。また、ライブラリー・ファイルを作成するためにリンク・エディット・スクリプトを実行しなければならないデータ・ソースもあります。

このタスクは次のデータ・ソースにのみ適用されます。

- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

手順:

ラッパー・ライブラリー・ファイルがフェデレーテッド・サーバー上にあるかどうかをチェックするには、次のようにします。

1. ラッパー・ライブラリーについて、ディレクトリー・パスのライブラリー・ファイルをチェックします。アクセスする各データ・ソースのフェデレーテッド・サーバーにライブラリー・ファイルが存在することを確認する必要があります。
2. ライブラリー・ファイルがディレクトリーに存在しない場合、ラッパー・ライブラリーを手動でデータ・ソース・クライアント・ソフトウェアにリンクする必要があります。

関連タスク:

- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 46 ページの『リンク・エディット・メッセージ・ファイルのチェック (UNIX)』
- 47 ページの『データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする』

関連資料:

- 45 ページの『ラッパー・ライブラリー・ファイル』
- 495 ページの『ライフ・サイエンス・ユーザー定義関数ライブラリー・ファイル』

ラッパー・ライブラリー・ファイル

ラッパー・ライブラリー・ファイルは、データ・ソースへのアクセスに必要です。データ・ソースの中には、DB2 Information Integrator のインストール時にライブラリー・ファイルがフェデレーテッド・サーバーに追加されるものがあります。また、ライブラリー・ファイルを作成するためにリンク・エディット・スクリプトを実行しなければならないデータ・ソースもあります。

ラッパー・ライブラリー・ファイルは、データ・ソースに対してラッパーを登録するときが必要です。

ラッパー・ライブラリー・ファイルがフェデレーテッド・サーバー上に存在することを確認する必要があります。アクセスするデータ・ソースごとに設定済みラッパー・ライブラリー・ファイルが存在していなければなりません。

ラッパー・ライブラリー・ファイルがフェデレーテッド・サーバー上に存在しない場合、リンク・エディット・スクリプトを手動で実行し、ライブラリー・ファイルを作成しなければなりません。

関連タスク:

- 44 ページの『ラッパー・ライブラリー・ファイルのチェック (UNIX)』

- 47 ページの『データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする』

関連資料:

- 120 ページの『BLAST ラッパーのライブラリー・ファイル』
- 82 ページの『BioRS ラッパーのライブラリー・ファイル』
- 179 ページの『DB2 ラッパーのライブラリー・ファイル』
- 195 ページの『Documentum ラッパーのライブラリー・ファイル』
- 221 ページの『Entrez ラッパーのライブラリー・ファイル』
- 246 ページの『Excel ラッパーのライブラリー・ファイル』
- 266 ページの『Extended Search ラッパーのライブラリー・ファイル』
- 294 ページの『HMMER ラッパー・ライブラリー・ファイル』
- 314 ページの『Informix ラッパー・ライブラリー・ファイル』
- 331 ページの『Microsoft SQL Server ラッパー・ライブラリー・ファイル』
- 345 ページの『ODBC ラッパー・ライブラリー・ファイル』
- 361 ページの『OLE DB ラッパー・ライブラリー・ファイル』
- 372 ページの『Oracle ラッパー・ライブラリー・ファイル』
- 387 ページの『Sybase ラッパー・ライブラリー・ファイル』
- 416 ページの『Teradata ラッパー・ライブラリー・ファイル』
- 401 ページの『表構造ファイル・ラッパー・ライブラリー・ファイル』
- 144 ページの『WebSphere Business Integration ラッパー・ライブラリー・ファイル』
- 435 ページの『Web サービス・ラッパー・ライブラリー・ファイル』
- 468 ページの『XML ラッパー・ライブラリー・ファイル』

リンク・エディット・メッセージ・ファイルのチェック (UNIX)

リンク・エディット・メッセージ・ファイルのチェックは、ラッパー・ライブラリーとデータ・ソース・クライアント・ソフトウェアの間のリンク・エディットを確認するという、より大きな作業の一部を成しています。

リンク・エディットに失敗すると、ライブラリー・ディレクトリー内のメッセージ・ファイルにエラーがリストされます。メッセージ・ファイルの存在は、リンク・エディットの失敗を意味するものではありません。リンク・エディットが成功しても、メッセージ・ファイルはライブラリー・ディレクトリーに存在します。

リンク・エディットが失敗したかどうかを判別するには、メッセージ・ファイルを開く必要があります。

このタスクは次のデータ・ソースでのみ必要です。

- Informix
- Microsoft SQL Server
- Oracle
- Sybase

• Teradata

手順:

リンク・エディットが失敗したかどうかを判別するには、リンク・エディット・メッセージ・ファイルを開きます。リンク・エディット・ファイルは、DB2 がインストールされているディレクトリーのサブディレクトリー lib または lib64 に存在します。リンク・エディット・メッセージ・ファイルの名前は、次の表にリストされています。

表 9. データ・ソース別のリンク・エディット・メッセージ・ファイル

データ・ソース	メッセージ・ファイル名
Informix	djxlinkInformix.out
Microsoft SQL Server	djxlinkMssql.out
Oracle	djxlinkOracle.out
Sybase	djxlinkSybase.out
Teradata	djxlinkTeradata.out

フェデレーテッド・サーバーのセットアップ時にリンクが失敗するのにはいくつかの理由があります。

- リンク・エディットが試行される前にデータ・ソース・クライアント・ソフトウェアがインストールされていない場合、リンク・エディットは失敗します。たとえば、DB2 サーバー・ソフトウェアをインストールする前に Informix クライアント・ソフトウェアをインストールしないと、リンク・エディットは失敗します。同様に、DB2 Information Integrator をインストールする前に Sybase Open Client ソフトウェアをインストールしないと、リンク・エディットは失敗します。このような状況では、リンクを手動で実行する必要があります。
- データ・ソース・クライアント・ソフトウェアのバージョンがサポートされていることを確認します。インストールしたデータ・ソース・クライアント・ソフトウェアのバージョンがサポートされていない場合、リンク・エディットは失敗します。サポートされているクライアント・バージョンをインストールし、その後リンクを手動で実行する必要があります。

関連タスク:

- 43 ページの『ラッパー・ライブラリー・ファイルのリンク・エディットの確認 (UNIX)』
- 44 ページの『ラッパー・ライブラリー・ファイルのチェック (UNIX)』
- 47 ページの『データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする』

データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを手動でリンクする

データ・ソース・クライアント・ソフトウェアへのラッパー・ライブラリーの手動リンクは、フェデレーテッド・サーバーのセットアップをチェックするという、より大きな作業の一部を成しています。

ライブラリー・ファイルがディレクトリー・パスに存在しない場合、ラッパー・ライブラリーを手動でリンクする必要があります。

このタスクは次のデータ・ソースにのみ適用されます。

- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

前提条件:

リンク・スクリプトを実行するためのルート権限が必要です。

djxlinkxxx スクリプトが母国語でメッセージを発行するには、少なくとも 1 つの DB2 インスタンスが存在しなければなりません。インスタンスが存在しなくてもスクリプトは機能します。ただし、エラー・メッセージを受け取ります。各エラー・メッセージは、db2djxmsg: Error retrieving message number で始まります。このエラー・メッセージの後、英語のメッセージが続きます。たとえば、次のようなメッセージです。

```
db2djxmsg: Error retrieving message number 9004
(return code -2029059891 from sqllogmsg).
Begin processing for wrapper: INFORMIX
INFORMIXDIR = /wsdb/v82/b1dsupp/AIX/informix2.81
db2djxmsg: Error retrieving message number 9015
(return code -2029059891 from sqllogmsg).
Library libdb2informixF.a was built successfully.
db2djxmsg: Error retrieving message number 9006
(return code -2029059891 from sqllogmsg).
End processing for wrapper: INFORMIX
```

手順:

データ・ソース・クライアント・ソフトウェアにラッパー・ライブラリーを短時間でリンクするには、次のようにします。

1. DB2 フェデレーテッド・サーバーでクライアント・ソフトウェアをインストールして構成します (必要な場合)。
2. リンクの実行に製品 CD を使用します。
 - Informix データ・ソースの場合、DB2 サーバー・インストールを再実行し、「標準」インストール・オプションを指定します。
 - Microsoft SQL Server、Oracle、Sybase、および Teradata データ・ソースの場合、DB2 Information Integrator インストールを再実行します。ランチパッドから「製品のインストール」をクリックし、ウィザードの指示に従います。
3. リンクを実行した後、ラッパー・ライブラリーの許可をチェックします。DB2 インスタンス所有者によるライブラリーの読み取りと実行が可能であることを確認します。

別の方法として、UNIX コマンド・プロンプトからリンク・エディット・スクリプトを実行することもできます。

1. UNIX コマンド・プロンプトを開きます。

2. アクセスするデータ・ソースごとに、リンク・エディット・スクリプトを実行します。

以下の表では、各データ・ソースのリンク・エディット・スクリプトの名前がリストされています。

表 10. データ・ソース別のリンク・スクリプト

データ・ソース	リンク・スクリプト名
Informix	djxlinkInformix
Microsoft SQL Server	djxlinkMssql
Oracle	djxlinkOracle
Sybase	djxlinkSybase
Teradata	djxlinkTeradata

たとえば、Informix データ・ソースにアクセスするようにフェデレーテッド・サーバーをセットアップする場合、UNIX コマンド・プロンプトから `djxlinkInformix` スクリプトを実行します。

```
djxlinkInformix
```

3. データ・ソースへのフェデレーテッド・アクセスを使用可能にするには、各 DB2 インスタンスで `db2iupdt` コマンドを発行します。
4. リンクを実行した後、ラッパー・ライブラリーの許可をチェックします。DB2 インスタンス所有者によるライブラリーの読み取りと実行が可能であることを確認します。

重要: DB2 Information Integrator によってサポートされるすべてのデータ・ソースに対してラッパー・ライブラリーの作成を試行する、`djxlink` という別のスクリプトもあります。`djxlink` スクリプトを実行し、一部のデータ・ソースのクライアント・ソフトウェアのみをインストールしている場合、インストールしていないそれぞれのデータ・ソースに関するエラー・メッセージを受け取ります。

関連タスク:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール (Windows)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール (UNIX)』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 49 ページの『FEDERATED パラメーターのチェック』
- 46 ページの『リンク・エディット・メッセージ・ファイルのチェック (UNIX)』

FEDERATED パラメーターのチェック

FEDERATED パラメーターのチェックは、フェデレーテッド・サーバーのセットアップをチェックするという、より大きな作業の一部を成しています。

フェデレーテッド・サーバーとデータベースにデータ・ソースを追加する前に、FEDERATED パラメーター設定をチェックする必要があります。

フェデレーテッド・サーバーのデータ・ソースへのアクセスを使用可能にするには、FEDERATED パラメーターを YES に設定する必要があります。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。

手順:

FEDERATED パラメーター設定をチェックするには、次のようにします。

1. 次の DB2 コマンドを発行して、すべてのパラメーターとその現行の設定を表示します。

```
GET DATABASE MANAGER CONFIGURATION
```

2. CONCENTRATOR パラメーター設定をチェックします。CONCENTRATOR パラメーターと FEDERATED パラメーターを同時に YES に設定することはできません。CONCENTRATOR パラメーターが YES に設定されている場合、設定を NO に変更してください。設定を変更するには次の DB2 コマンドを発行します。

```
UPDATE DATABASE MANAGER CONFIGURATION USING CONCENTRATOR NO
```

3. FEDERATED パラメーター設定をチェックします。FEDERATED パラメーターが NO に設定されている場合、設定を YES に変更してください。設定を変更するには次の DB2 コマンドを発行します。

```
UPDATE DATABASE MANAGER CONFIGURATION USING FEDERATED YES
```

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』

関連タスク:

- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

第 4 章 フェデレーテッド・データベースの作成

データ・ソースにアクセスできるようにフェデレーテッド・サーバーを構成する前に、フェデレーテッド・サーバーとして使用するデータベースを作成する必要があります。

フェデレーテッド・データベースの各国語に関する考慮事項

多くのリレーショナル・データ・ソースで、ラッパーはデータ・ソースへ接続する際に以下のタスクを実行します。

1. フェデレーテッド・データベースのコード・ページおよびテリトリーを判別します。
2. そのコード・ページおよびテリトリーをデータ・ソースのクライアント・ロケール名にマップします。
3. データ・ソースに応じて、環境変数を設定するか、またはデータ・ソース API を呼び出して、クライアント・ロケールをデータ・ソースに指示します。
4. それから、データ・ソースがリモート・データベースのコード・ページとフェデレーテッド・データベースのコード・ページの間で文字データを変換します。

たとえば、フェデレーテッド・データベースはコード・ページ 819、テリトリー US を使用します。それに相当する Oracle クライアント・ロケールは American_America.WE8ISO8859P1 です。ラッパーは NLS_LANG 変数を Oracle クライアント・ロケール値に設定します。データが Oracle データベースからラッパーへ送信される場合、Oracle データベースがデータをコード・セット American_America.WE8ISO8859P1 からコード・ページ 819 へ変換します。データが Oracle データベースからラッパーへ送信される場合、Oracle サーバーまたはクライアントがデータを Oracle データベースのコード・ページからコード・ページ 819 へ変換します。データがラッパーから Oracle データベースへ送信される場合、Oracle サーバーまたはクライアントがデータをコード・ページ 819 から Oracle データベースのコード・ページへ変換します。

コード・ページの変換を実行しないリレーショナル・データ・ソースの場合、一部のラッパーが変換を実行します。

データ・ソースと同じコード・セット、テリトリー、および照合シーケンスを使用するようにフェデレーテッド・データベースを定義すると、パフォーマンスを向上させることができます。データ・ソースと同じコード・セット、テリトリー、および照合シーケンスを使用するようにフェデレーテッド・データベースを定義する場合、コード・ページの変換は必要なくなります。大量の文字データを変換する場合、同じ各国語設定を使用することにより、パフォーマンスを向上させることができます。

フェデレーテッド・データベースでコード・ページおよびテリトリーを指定するには、CREATE DATABASE ステートメントで USING CODESET および TERRITORY パラメーターを使用します。

関連概念:

- 55 ページの『フェデレーテッド・システムの Unicode サポート』
- 52 ページの『フェデレーテッド・システムの照合シーケンス』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 54 ページの『フェデレーテッド・データベース照合シーケンスの設定』

フェデレーテッド・システムの照合シーケンス

フェデレーテッド・サーバーが照会を受け取ると、DB2[®] SQL コンパイラーはグローバル・カタログ内の情報およびデータ・ソース・ラッパーを検討し、照会の処理に役立てます。SQL コンパイラー処理の一部として、照会オプティマイザーは照会を分析します。コンパイラーは、アクセス・プランと呼ばれる、照会を処理するための代替ストラテジーを作成します。アクセス・プランは、照会について次のような処理方法を要求します。

- 照会をデータ・ソースが処理する
- 照会をフェデレーテッド・サーバーが処理する
- 照会の一部をデータ・ソースが処理し、一部をフェデレーテッド・サーバーが処理する

一般に、データ・ソースで文字ソートおよび文字比較を実行すると、パフォーマンスは向上します。

照会が文字ソートおよび文字比較を必要とする場合、SQL コンパイラーは照合シーケンス情報を使用して、使用するアクセス・プランを決定します。

デフォルトでは、フェデレーテッド・データベース照合シーケンスには大文字小文字の区別があります。しかし、大文字小文字を区別しない照合シーケンスを使用するフェデレーテッド・データ・ソースもあります。フェデレーテッド・データベースとデータ・ソースの照合シーケンスが異なる場合、照会の結果は異なる場合があります。操作が文字ソートの場合、同じデータが戻されますが、結果の順序は異なります。操作が文字比較の場合、戻される結果が異なる場合があります。

ソートおよび比較が処理される場所は、いくつかの要因によって決まります。

- フェデレーテッド・データベースの照合シーケンスがデータ・ソース側の照合シーケンスと同じ場合、文字ソートおよび文字比較はデータ・ソース側で行うことができます。照会オプティマイザーは、ローカル操作とリモート操作のどちらがより効率的に照会を完了できるかを判断することができます。データ・ソースによるすべてのタイプの文字比較および文字ソートは、それらのアクションがフェデレーテッド・データベースによって実行されるかのように、同じ結果になることが想定されます。
- フェデレーテッド・データベースの照合シーケンスがデータ・ソース側の照合シーケンスと異なるが、データ・ソース側の照合シーケンスに大文字小文字の区別がある場合、文字ソートまたは文字比較は、フェデレーテッド・データベースで行われます。データ・ソースは、WHERE=、DISTINCT、および GROUP BY 操

作の文字データに関して同じ結果になります。しかし、より大またはより小の述部を持つ ORDER BY や WHERE のような他の操作の場合、文字データに関して異なる結果になります。

- フェデレーテッド・データベースの照合シーケンスがデータ・ソース側の照合シーケンスと異なるが、データ・ソース側の照合シーケンスが大文字小文字を区別しない場合、文字ソートまたは文字比較は、フェデレーテッド・データベースで行われます。データ・ソースは大文字小文字を同等のものとみなし、要求された操作で大文字が指定されたか小文字が指定されたかに関係なく、それらの両方を結果に組み込むことが想定されます。WHERE=、より大またはより小の述部を持つ WHERE、ORDER BY、DISTINCT、および GROUP BY 操作は、データ・ソースにプッシュダウンされません。

たとえば、大文字小文字を区別しないデータ・ソースは、“S” と “s” のどちらにも同じウェイトを割り当てます。英語コード・ページの大文字小文字を区別しないデータ・ソースでは、**STEWART**、**SteWArT**、**stewart** はすべて同じと見なされます。ただし、大文字小文字の区別のある照合シーケンスが使用されると、文字に対して異なるウェイトが割り当てられます。照合シーケンスの大文字小文字の区別によって、文字ソートおよび文字比較の結果セットは異なります。

フェデレーテッド・データベースとデータ・ソースの照合シーケンスが異なる場合、フェデレーテッド・サーバーはデータを取り出し、文字ソートおよび文字比較はローカルで実行されます。これらのタスクをローカルで実行する理由は、DB2 ユーザーは、フェデレーテッド・サーバーに定義された照合シーケンスにしたがって並べられた照会結果を見ることを期待しているからです。データをローカルで並び替えることにより、結果セットの一貫性が DB2 ユーザーに保証されます。データをローカルでソートおよび比較するために検索すると、通常はパフォーマンスが低下します。

データ・ソース側の照合シーケンスで並べられた文字データを見る必要がある場合は、照会をパススルー・セッションでサブミットすることができます。

データ・ソースとフェデレーテッド・データベースの照合シーケンスが同じかどうかを判断するには、以下の要因を検討してください。

コード・ページ

フェデレーテッド・サーバーおよびデータ・ソースによって使用される、ASCII や EBCDIC などのコード・ページ体系は結果に影響を与えます。

各国語サポート (NLS)

照合シーケンスは、サーバーでサポートされる言語に関係しています。ご使用のオペレーティング・システムの DB2 NLS 情報とデータ・ソース側の NLS 情報を比較してください。

データ・ソースの特性

データ・ソースによっては、大文字小文字を区別しない照合シーケンスを使用して作成されており、その場合、順序に関係する操作では、DB2 とは異なる結果になることがあります。

カスタマイズ

あるデータ・ソースでは、照合シーケンスに複数のオプションを提供したり、照合シーケンスをカスタマイズできたりします。

照合シーケンスの設定のためのオプションがいくつかあります。以下を行うことができます。

- フェデレーテッド・データベースの作成時に照合シーケンスを設定する。
- データ・ソースのサーバー定義の作成時に `COLLATING_SEQUENCE` オプションを設定する。このオプションは、リレーショナル・データ・ソースでのみ使用できます。

関連概念:

- 55 ページの『フェデレーテッド・システムの Unicode サポート』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 54 ページの『フェデレーテッド・データベース照合シーケンスの設定』

関連資料:

- 51 ページの『フェデレーテッド・データベースの各国語に関する考慮事項』

フェデレーテッド・データベース照合シーケンスの設定

管理者は、データ・ソースの照合シーケンスに一致する特定の照合シーケンスを持つフェデレーテッド・データベースを作成することができます。

フェデレーテッド・データベースの照合シーケンスは、`CREATE DATABASE API`の一部として設定します。この API を通して、次のいずれかのシーケンスを指定できます。

- 一致シーケンス
- システム・シーケンス (データベースをサポートするオペレーティング・システムが使用するシーケンス)
- カスタマイズ・シーケンス (DB2 UDB が用意する事前定義またはユーザー定義の事前定義)

手順:

フェデレーテッド・データベースの照合シーケンスを指定するには、`CREATE DATABASE` ステートメントで `COLLATE USING` パラメーターを使用します。

リレーショナル・データ・ソースの場合、フェデレーテッド・データベースとデータ・ソースが同じ照合シーケンスを使用する場合、`COLLATING_SEQUENCE` サーバー・オプションを 'Y' に設定する必要があります。`COLLATING_SEQUENCE` サーバー・オプションを 'Y' に設定すると、フェデレーテッド・サーバーにフェデレーテッド・データベースとデータ・ソースの照合シーケンスが一致していることを知らせます。リレーショナル・データ・ソースに対してサーバー定義を作成するときに `COLLATING_SEQUENCE` サーバーを設定します。

`COLLATING_SEQUENCE` サーバー・オプションをサポートするリレーショナル・データ・ソースは次のとおりです。

- DB2 ファミリー
- Informix
- Microsoft SQL Server

- ODBC
- OLE DB
- Oracle
- Sybase
- Teradata

関連概念:

- 55 ページの『フェデレーテッド・システムの Unicode サポート』
- 52 ページの『フェデレーテッド・システムの照合シーケンス』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』

関連資料:

- 51 ページの『フェデレーテッド・データベースの各国語に関する考慮事項』

フェデレーテッド・システムの Unicode サポート

リレーショナル・ラッパーと非リレーショナルのラッパー、およびユーザー定義関数は、Unicode データベース (UTF-8 データベース) 上で実行できます。Unicode データベースは、プラットフォームに依存しないフェデレーテッド・サーバー環境を提供します。Unicode データベースは、種々のデータ・ソース上のさまざまなコード・ページ内に保管されたデータを取り扱うことができます。

Unicode をサポートするラッパーおよびユーザー定義関数は、以下のとおりです。

- リレーショナル・ラッパー
 - DRDA[®]
 - Informix[®]
 - MS SQL Server
 - ODBC
 - OLE DB
 - Oracle
 - Sybase
 - Teradata
- 非リレーショナル・ラッパーおよびユーザー定義関数
 - BioRS ラッパー
 - BLASTwrapper
 - Documentum ラッパー
 - Entrez ラッパー
 - Excel ラッパー
 - HMMER ラッパー
 - IBM[®] Lotus[®] Extended Search ラッパー
 - KEGG ユーザー定義関数
 - MQ ユーザー定義関数
 - 表構造ファイルのラッパー
 - Web サービス・ユーザー定義関数
 - Web サービス・ラッパー

- WebSphere® Business Integration ラッパー
- XML ラッパー

図 5 で、ある会社は、異なる国々に事業所を持っています。各事業所は、独自のコード・ページの独自のデータベースに顧客データを保管しています。Microsoft® SQL Server データベースは、データをコード・ページ A で保管します。Oracle データベースはデータをコード・ページ B で保管します。コード・ページ A とコード・ページ B は、異なるテリトリーにあります。異なるテリトリーにあるデータをフェデレーテッドするため、この会社ではフェデレーテッド・データベースのコード・ページを Unicode に設定することができます。そうすれば、この会社では、テリトリーに関係なく表を結合して合計購入注文数を参照できるようになります。

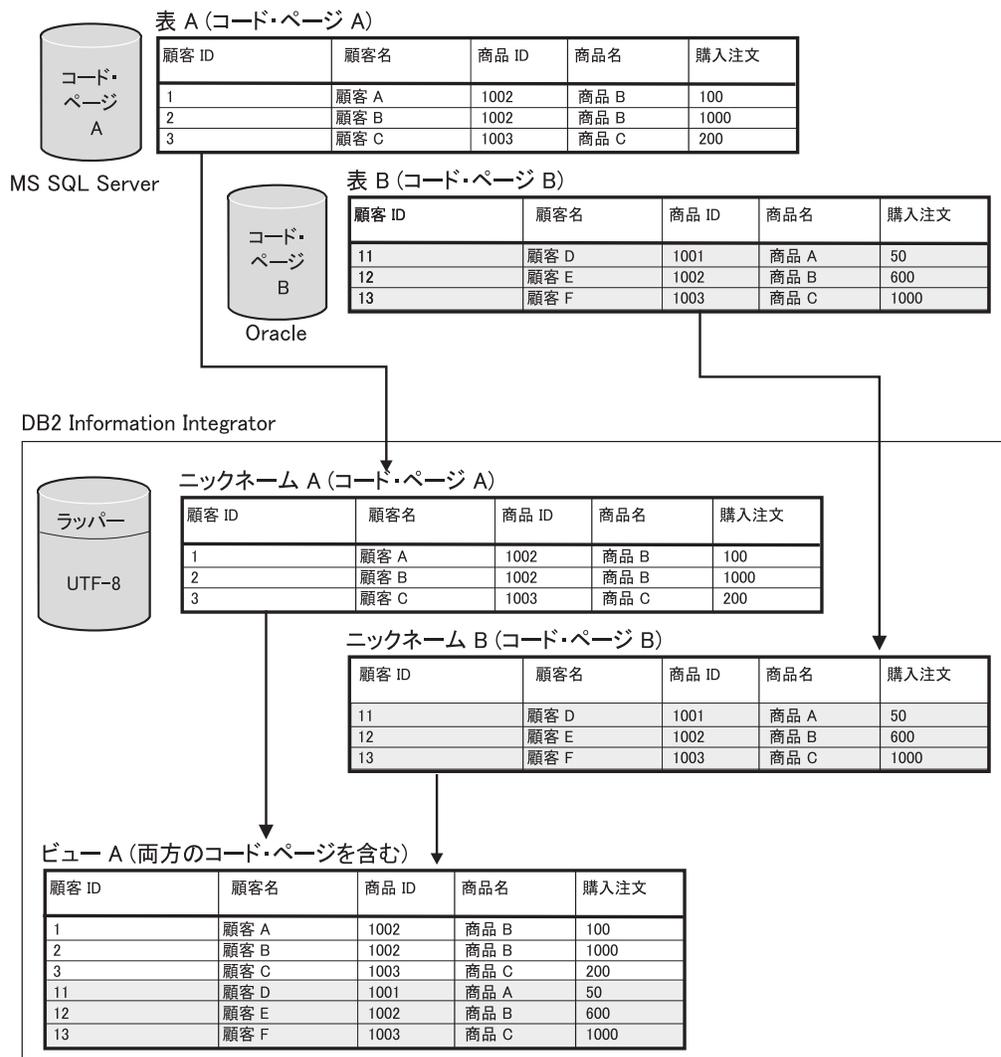


図 5. Unicode の例

関連タスク:

- 「フェデレーテッド・システム・ガイド」の『Microsoft SQL Server データ・ソースおよび ODBC データ・ソースの Unicode サポートのためのクライアント・コード・ページの指定 - 例』

- ・ 「フェデレーテッド・システム・ガイド」の『表構造ファイル・データ・ソースの Unicode サポートのためのファイル・コード・ページの指定』

関連資料:

- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - NET8 ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『MSSQL および ODBC ラッパーの CODEPAGE オプションでサポートされる Unicode コード・ページ』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの逆方向データ・タイプ・マッピング - NET8 ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - Sybase ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの逆方向データ・タイプ・マッピング - Sybase ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - ODBC ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの逆方向データ・タイプ・マッピング - ODBC ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - Microsoft SQL Server ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの逆方向データ・タイプ・マッピング - Microsoft SQL Server ラッパー』
- ・ 「フェデレーテッド・システム・ガイド」の『表構造ファイル・データ・ソースの Unicode サポートのためのファイル・コード・ページの指定 - 例』

フェデレーテッド・データベースの作成

フェデレーテッド・サーバーをセットアップした後、DB2 インスタンス所有者は、フェデレーテッド・データベースとして機能するフェデレーテッド・サーバー・インスタンス上に DB2 データベースを作成する必要があります。

推奨事項: 接続先のリモート・データ・ソースが異なるまたは非互換コード・ページを使用している場合、フェデレーテッド・データベースを Unicode データベースとして定義してください。フェデレーテッド・データベースを Unicode データベースとして定義するには、CREATE DATABASE ステートメントに USING CODESET UTF-8 を指定します。

このステップは、データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成する前に完了しておく必要があります。

前提条件:

- ・ DB2 データベースを作成するための SYSADM または SYSCTRL 権限。
- ・ DB2 Information Integrator をフェデレーテッド・サーバーとして機能するサーバー上にインストールしておく必要があります。
- ・ フェデレーテッド・データベースの作成時に照合シーケンスを指定するかどうかを決定します。

|

手順:

| フェデレーテッド・データベースは、DB2 コントロール・センターまたは DB2 コマンド行から作成できます。

| DB2 コントロール・センターからこのタスクを行うには、以下のようになります。

- | 1. 「データベース」フォルダーを右クリックし、「作成」->「データベース (ウィザードを使用)」をクリックします。「データベースの作成」ウィザードがオープンします。
- | 2. ウィザードのステップを最後まで行います。

| この作業を DB2 コマンド行から行うには、CREATE DATABASE ステートメントを発行します。例:

|

```
CREATE DATABASE federated
```

| このコマンドにより、以下の作業が行われます。

- | • 新規データベースの初期化
- | • 3 つの初期表スペースの作成
- | • システム表の作成
- | • リカバリー・ログの割り振り

| DB2 インスタンスが複数のパーティション構成を使用している場合、CREATE DATABASE コマンドは、db2nodes.cfg ファイルにリストされているすべてのパーティションに影響を与えます。このコマンドの発行元データベース・パーティションは、新規データベース用のカタログ・パーティションになります。

|

関連概念:

- | • 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator のインストール - 概要』
- | • 61 ページの『データ・ソースの構成の近道』

|

関連タスク:

- | • 54 ページの『フェデレーテッド・データベース照合シーケンスの設定』

第 3 部 データ・ソース

第 5 章 データ・ソースへのアクセスの構成の概要

以下の各セクションは、フェデレーテッド・サーバーおよびフェデレーテッド・データベースを構成して、データ・ソースにアクセスできるようにするための簡単なガイドになっています。

- 構成ステップをすばやく実行するのに必要な基本ステップについて説明します。
- データ・ソースの構成を微調整するための、いくつかの必要に応じたオプションのステップを簡単に説明します。

それぞれのデータ・ソースごとに、構成に関する別々の章があります。

データ・ソースの構成の近道

データ・ソースへのアクセスを構成するために必要なステップのほとんどは、DB2[®] コントロール・センターを使用して行うことができます。DB2 コマンド・センターは、コマンド行を必要とするステップで使用してください。これらのグラフィカル・ユーザー・インターフェースを切り替えて、データ・ソースへのアクセスを迅速に構成できます。

データ・ソースへのアクセスを構成する前にフェデレーテッド・サーバーが適切にセットアップされていることを確認してください。

データ・ソースへアクセスするためにフェデレーテッド・サーバーを構成するステップは、データ・ソースに関係なく類似しています。基本的なステップとお勧めするインターフェースを以下に示します。

表 11. お勧めするインターフェースおよび構成ステップ

構成ステップ	推奨インターフェース	注
1. データ・ソース用のフェデレーテッド・サーバーを準備します。	クライアント構成アシスタント	一部のデータ・ソースでのみ必要です。このステップでは、ソフトウェアのインストール、ファイルの構成、または設定の検査が必要になる場合があります。
2. 必要な環境変数を設定します。	DB2 コントロール・センター	環境変数が必要なものを以下に示します。 <ul style="list-style-type: none">• Documentum• Informix[®]• Microsoft[®] SQL Server• Oracle• Sybase• Teradata

表 11. お勧めするインターフェースおよび構成ステップ (続き)

構成ステップ	推奨インターフェース	注
2. ラッパーを登録します。	DB2 コントロール・センターの「フェデレーテッド・オブジェクト (Federated Objects)」ウィザード。	ラッパーは、アクセスするデータ・ソースごとに必要です。
3. サーバー定義を登録します。	DB2 コントロール・センターの「フェデレーテッド・オブジェクト (Federated Objects)」ウィザード。	サーバー定義はラッパーと関連付けられ、ニックネームを登録する際に使用されます。
4. ユーザー・マッピングを作成します。	DB2 コントロール・センターの「フェデレーテッド・オブジェクト (Federated Objects)」ウィザード。	一部のデータ・ソースでのみ必要です。 ユーザー・マッピングと関連付けられているリモート・パスワードを SYSCAT.USEROPTIONS カタログ・ビューから検索しようとする、リモート・パスワードの値は暗号化されて表示されます。
5. データ・ソース・サーバーへの接続をテストします。	DB2 コマンド・センター	一部のデータ・ソースでのみ必要です。
6. ニックネームを作成します。	DB2 コントロール・センターの「フェデレーテッド・オブジェクト (Federated Objects)」ウィザード。	ニックネームは、アクセスするデータ・ソース・オブジェクトごとに必要です。

関連概念:

- 74 ページの『オプションの構成ステップ』

関連タスク:

- 63 ページの『DB2 UDB コントロール・センターを使用したフェデレーテッド・サーバーへのデータ・ソースの追加』
- 68 ページの『データ・ソースのラッパーの登録』
- 69 ページの『データ・ソースのサーバー定義の登録』
- 70 ページの『データ・ソースのユーザー・マッピングの登録』
- 71 ページの『データ・ソースのニックネームの登録』
- 65 ページの『データ・ソース環境変数の設定』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

DB2 UDB コントロール・センターを使用したフェデレーテッド・サーバーへのデータ・ソースの追加

データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

手順:

DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。ウィザードを開始するには、「フェデレーテッド・データベース・オブジェクト」フォルダーを右マウス・ボタンでクリックし、フェデレーテッド・オブジェクトの作成をクリックします。

フェデレーテッド・サーバーを構成するのに必要なステップは、データ・ソースごとに異なります。

「アクション出力」ウィンドウを使用して、データ・ソースへアクセスするように複数のフェデレーテッド・サーバーを構成できます。

関連タスク:

- 64 ページの『データ・ソースへアクセスするための複数のフェデレーテッド・サーバーの構成』
- 400 ページの『フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加』
- 191 ページの『フェデレーテッド・サーバーへの Documentum データ・ソースの追加』
- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』
- 111 ページの『フェデレーテッド・サーバーへの BLAST データ・ソースの追加』
- 218 ページの『フェデレーテッド・サーバーへの Entrez データ・ソースの追加』
- 409 ページの『フェデレーテッド・サーバーへの Teradata データ・ソースの追加』
- 78 ページの『フェデレーテッド・サーバーへの BioRS データ・ソースの追加』
- 175 ページの『フェデレーテッド・サーバーへの DB2 ファミリー・データ・ソースの追加』
- 264 ページの『フェデレーテッド・サーバーへの Extended Search データ・ソースの追加』
- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』
- 307 ページの『フェデレーテッド・サーバーへの Informix データ・ソースの追加』
- 325 ページの『フェデレーテッド・サーバーへの Microsoft SQL Server データ・ソースの追加』
- 341 ページの『フェデレーテッド・サーバーへの ODBC データ・ソースの追加』

- 359 ページの『フェデレーテッド・サーバーへの OLE DB データ・ソースの追加』
- 365 ページの『フェデレーテッド・サーバーへの Oracle データ・ソースの追加』
- 381 ページの『フェデレーテッド・サーバーへの Sybase データ・ソースの追加』
- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』

データ・ソースへアクセスするための複数のフェデレーテッド・サーバーの構成

フェデレーテッド・システムを複数のフェデレーテッド・サーバーで構成することができます。各フェデレーテッド・サーバーを個別に構成するのではなく、DB2 コントロール・センターを使用してフェデレーテッド・サーバーを構成することによって、時間を節約できます。最初のサーバーを構成する際に、フェデレーテッド・オブジェクトの作成時に発行される DDL ステートメントが、「アクション出力」ウィンドウでキャプチャーされます。これらのステートメントを再利用または変更し、ステートメントを適用することにより、追加のフェデレーテッド・サーバーをすぐに構成できます。

現行セッションでは、「アクション出力」ウィンドウはアクティブのままです。「アクション出力」ウィンドウをクローズしても、現行セッションの DDL ステートメントは引き続き「アクション出力」ウィンドウに保管されます。ただし、DB2 コントロール・センターをクローズすると、現行セッションからのすべての DDL ステートメントが「アクション出力」ウィンドウから除去されます。

前提条件:

- フェデレーテッド・サーバーとしての役割を果たすサーバーに、DB2 Information Integrator がインストール済み。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

データ・ソースへアクセスするための複数のフェデレーテッド・サーバーを構成するには、以下を行います。

1. DB2 コントロール・センターを使用して、アクセスするデータ・ソース用に最初のフェデレーテッド・サーバーを構成します。これにより、各 DDL ステートメントがキャプチャーされます。
2. 「アクション出力」ウィンドウに「アクション出力」ページが表示されます。

「アクション出力」ウィンドウをクローズする場合、「フェデレーテッド・データベース・オブジェクト」フォルダーを右マウス・ボタンでクリックし、「アクションの表示」をクリックして、「アクション出力」ウィンドウをオープンします。

3. 他のフェデレーテッド・サーバーで使用しない DDL ステートメントをすべて削除します。ステートメントを削除するには、そのステートメントを右マウス・ボタンでクリックし、「**除去**」をクリックします。たとえば、「**アクション出力**」ページで状況列に「**失敗**」が表示されているすべてのステートメントを削除することもできます。
4. 以下のように、他のフェデレーテッド・サーバーで使用するステートメントを「**コマンド・エディター**」ページにコピーします。
 - a. コピーするステートメントを選択します。複数のステートメントを選択するには、Ctrl キーを使用します。
 - b. 選択したステートメントを右マウス・ボタンでクリックし、「**コマンド・エディターにコピー**」をクリックします。「**コマンド・エディター**」ページがオープンします。
5. 他のフェデレーテッド・サーバーで使用する、「**コマンド・エディター**」ページ内のすべての DDL ステートメントを変更します。たとえば、ローカル・スキーマを指定するすべてのステートメントを変更することもできます。

ユーザー・マッピングを変更して、パスワードを指定しなければなりません。CREATE USER MAPPING ステートメントの DDL が「**アクション出力**」ウィンドウでキャプチャーされるときには、パスワードはアスタリスクで隠されます。このアスタリスクを適切なパスワードに置き換えなければなりません。

6. 次のフェデレーテッド・サーバーで DDL ステートメントを発行します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』

データ・ソース環境変数の設定

データ・ソース環境変数の設定

DB2 Information Integrator をインストールする場合、インストール・プロセスでは、いくつかのデータ・ソースで必要となる環境変数の設定が試行されます。DB2 Information Integrator をインストールする前にフェデレーテッド・サーバーをクライアント・ソフトウェアをインストールしていなかった場合など、環境変数を設定できない場合もあります。

データ・ソースをフェデレーテッド・サーバーに追加するためのステップに従えば、環境変数の検査、およびそれらの設定 (必要な場合) を行えます。

- DB2 コントロール・センターを使用して、データ・ソースをフェデレーテッド・サーバーに追加する場合、環境変数に関する要件が自動的に検査されます。ラッパーを作成または変更する際に、環境変数を設定できます。
- DB2 コマンド行を使用して、データ・ソースをフェデレーテッド・サーバーに追加する場合、環境変数の設定は手動で行わなければなりません。

以下のデータ・ソースでは、環境変数の設定が必要です。

- Documentum
- Informix
- Microsoft SQL Server
- Oracle
- Sybase
- Teradata

前提条件:

このタスクはシステム管理者が実行しなければなりません。

制約事項:

次のトピックを参照してください。『db2dj.ini ファイルの制約事項』

手順:

環境変数を検査するためのステップは、データ・ソースごとに異なります。

関連概念:

- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 193 ページの『Documentum 環境変数の設定』
- 68 ページの『データ・ソースのラッパーの登録』
- 309 ページの『Informix 環境変数の設定』
- 328 ページの『Microsoft SQL Server 環境変数の設定』
- 366 ページの『Oracle 環境変数の設定』
- 382 ページの『Sybase 環境変数の設定』
- 412 ページの『Teradata 環境変数の設定』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』

db2dj.ini ファイルの制約事項

db2dj.ini ファイルには、以下の制約事項が適用されます。

- 項目の形式は *evname=value* でなければならない。

evname は環境変数の名前、*value* はその値です。

- 環境変数の名前の最大長は 255 バイト。
- 環境変数の値の最大長は 765 バイト。
- ファイル内の行の最大長はすべて 1021 バイト。その長さを超えるデータは無視されます。

関連タスク:

- 193 ページの『Documentum 環境変数の設定』
- 65 ページの『データ・ソース環境変数の設定』
- 309 ページの『Informix 環境変数の設定』

- 328 ページの『Microsoft SQL Server 環境変数の設定』
- 366 ページの『Oracle 環境変数の設定』
- 382 ページの『Sybase 環境変数の設定』
- 412 ページの『Teradata 環境変数の設定』

複数パーティション・インスタンス構成での環境変数の適用

フェデレーテッド・サーバー・インスタンスが複数パーティション構成を持つ場合、すべてのパーティションにデータ・ソース環境変数を適用する必要があります。

このステップは、フェデレーテッド・サーバーが複数パーティション・インスタンス構成を持つ場合にのみ必要です。

db2dj.ini ファイルには、データ・ソース環境変数が入っています。このファイルは、DB2 Information Integrator のインストール時にフェデレーテッド・サーバーに追加されます。

同じ db2dj.ini ファイルのコピーを複数パーティション・インスタンス構成のすべてのパーティションに追加する必要があります。ファイルのデフォルト名は db2dj.ini ファイルです。UNIX フェデレーテッド・サーバーでは、db2dj.ini ファイルのデフォルト・パスは、INSTHOME/sql1lib/cfg です。INSTHOME は、インスタンス所有者のホーム・ディレクトリです。Windows フェデレーテッド・サーバーでは、ファイルへのデフォルト・パスは、x:%SQLLIB%cfg です。x:%SQLLIB は、DB2PATH レジストリー変数または環境変数で指定されたドライブとディレクトリです。

db2dj.ini ファイルのデフォルト・パスとファイル名は、DB2_DJ_INI レジストリー変数によってオーバーライドされる可能性があります。

すべてのパーティションまたはパーティションのサブセットに DB2_DJ_INI レジストリー変数を適用できます。

手順:

フェデレーテッド・サーバー上の該当するパーティションに DB2_DJ_INI レジストリー変数を適用するには、**db2set** コマンドを使用します。**db2set** コマンドは、DB2 プロファイル変数を表示、設定、または除去します。

使用する **db2set** コマンドの構文は、データベース・システムの構造によって異なります。

- このインスタンス内のすべてのデータベース・パーティションに DB2_DJ_INI レジストリー変数を適用するには、次のコマンドを発行します。

```
db2set -g DB2_DJ_INI=$HOME/sql1lib/cfg/my_db2dj.ini
```

- 現行パーティションにのみ DB2_DJ_INI レジストリー変数を適用するには、次のコマンドを発行します。

```
db2set DB2_DJ_INI=$HOME/sql1lib/cfg/my_db2dj.ini
```

- 特定のパーティションに DB2_DJ_INI レジストリー変数を適用するには、次のコマンドを発行します。

```
db2set -i INSTANCEX 3 DB2_DJ_INI=$HOME/sqllib/cfg/partition3.ini
```

INSTANCEX

インスタンスの名前。

3 db2nodes.cfg ファイルにリストされるパーティション番号。

partition3.ini

db2dj.ini ファイルの変更バージョンおよびリネーム・バージョン。

重要: DB2_DJ_INI レジストリー変数を設定する際は、絶対パスに設定する必要があります。FEDERATED パラメーターが YES に設定され、DB2_DJ_INI レジストリー変数が相対パスに設定されると、DB2 Universal Database エンジンでは開始されません。

関連タスク:

- 65 ページの『データ・ソース環境変数の設定』

データ・ソースのラッパーの登録

アクセスするデータ・ソースのタイプごとに、ラッパーを 1 つ登録します。3 つの DB2 for z/OS データベース表、1 つの DB2 for iSeries 表、2 つの BLAST 検索タイプにアクセスするには、2 つのラッパーを作成する必要があります。DB2 データベース用に 1 つのラッパー、BLAST 検索タイプ用に 1 つのラッパーを登録する必要があります。

フェデレーテッド・データベースにこれらのラッパーが登録されると、それらのラッパーを使用して、これらのデータ・ソースのその他のオブジェクトにアクセスすることができます。たとえば、DRDA ラッパーを使用して、すべての DB2 ファミリーのデータ・ソース・オブジェクト (DB2 for Linux、UNIX、Windows、DB2 for z/OS and OS/390、DB2 for iSeries、および DB2 Server for VM and VSE) からのデータ・ソースへアクセスすることができます。

手順:

ラッパーは、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- DB2 コントロール・センターからこのタスクを行うには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、または「**フェデレーテッド・オブジェクト (Federated Objects)**」フォルダーを右マウス・ボタンでクリックし、「ラッパー作成」をクリックします。
- このタスクを DB2 コマンド行から行うには、CREATE WRAPPER ステートメントを使用します。

関連タスク:

- 400 ページの『表構造ファイル・ラッパーの登録』
- 194 ページの『Documentum ラッパーの登録』
- 245 ページの『Excel ラッパーの登録』
- 119 ページの『BLAST ラッパーの登録』
- 467 ページの『XML ラッパーの登録』

- 220 ページの『Entrez ラッパーの登録』
- 415 ページの『Teradata ラッパーの登録』
- 81 ページの『BioRS ラッパーの登録』
- 178 ページの『DB2 ラッパーの登録』
- 265 ページの『Extended Search ラッパーの登録』
- 293 ページの『HMMER ラッパーの登録』
- 313 ページの『Informix ラッパーの登録』
- 330 ページの『Microsoft SQL Server ラッパーの登録』
- 344 ページの『ODBC ラッパーの登録』
- 360 ページの『OLE DB ラッパーの登録』
- 371 ページの『Oracle ラッパーの登録』
- 386 ページの『Sybase ラッパーの登録』
- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 434 ページの『Web サービス・ラッパーの登録』

データ・ソースのサーバー定義の登録

サーバー定義の目的は、データ・ソースによって異なります。

通常、リレーショナル・データ・ソースのサーバー定義はリモート・データベース、データベース・パーティション、またはノードを表します。非リレーショナル・データ・ソースの場合、サーバー定義の中には、検索タイプとデーモン、Web サイト、または Web サーバーへマップするものがあります。他の非リレーショナル・データ・ソースの場合、フェデレーションで必要となる場合にのみサーバー定義が作成されます。

ニックネームを作成するすべてのデータ・ソース・オブジェクトを、特定のサーバー定義と関連付けなければなりません。

いくつかのデータ・ソースでは、サーバー定義を登録する際にノードを指定しなければなりません。ノードの概念はデータ・ソースによって異なります。リレーショナル・データ・ソースの場合、ノードはデータ・ソースのサーバー・インスタンスを反映します。

手順:

サーバー定義は、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- このタスクを DB2 コントロール・センターから実行するには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、あるいは「**サーバー定義 (Server Definitions)**」フォルダーを右マウス・ボタンでクリックして、「**作成**」をクリックします。
- このタスクを DB2 コマンド行から行うには、CREATE SERVER ステートメントを使用します。

関連タスク:

- 402 ページの『表構造ファイルのサーバーの登録』
- 195 ページの『Documentum データ・ソースのサーバーの登録』
- 247 ページの『Excel データ・ソースのサーバーの登録』
- 121 ページの『BLAST データ・ソースのサーバーの登録』
- 469 ページの『XML データ・ソースのサーバーの登録』
- 222 ページの『Entrez データ・ソースのサーバーの登録』
- 416 ページの『Teradata データ・ソースのサーバー定義の登録』
- 82 ページの『BioRS データ・ソースのサーバー定義の登録』
- 180 ページの『DB2 データ・ソースのサーバー定義の登録』
- 266 ページの『Extended Search データ・ソースのサーバーの登録』
- 294 ページの『HMMER データ・ソースのサーバー定義の登録』
- 315 ページの『Informix データ・ソースのサーバー定義の登録』
- 332 ページの『Microsoft SQL Server データ・ソースのサーバー定義の登録』
- 346 ページの『ODBC データ・ソースのサーバー定義の登録』
- 361 ページの『OLE DB データ・ソースのサーバー定義の登録』
- 373 ページの『Oracle データ・ソースのサーバー定義の登録』
- 387 ページの『Sybase データ・ソースのサーバー定義の登録』
- 145 ページの『ビジネス・アプリケーション・データ・ソースのサーバー定義の登録』
- 435 ページの『Web サービス・データ・ソースのサーバー定義の登録』

データ・ソースのユーザー・マッピングの登録

同じデータ・ソースの場合、フェデレーテッド・サーバーの許可 ID と、データ・ソースのユーザー ID およびパスワードの間の関連を定義しなければなりません。フェデレーテッド・システムを使用する、各ユーザー ID のユーザー・マッピングを作成して、分散要求を送信します。

手順:

ユーザー・マッピングは、DB2 コントロール・センターまたは DB2 コマンド行から作成できます。

- DB2 コントロール・センターからこのタスクを行うには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、または「ユーザー・マッピング」フォルダーを右マウス・ボタンでクリックし、「作成」をクリックします。
- このタスクを DB2 コマンド行から行うには、CREATE USER MAPPING ステートメントを使用します。

関連タスク:

- 196 ページの『Documentum データ・ソースのユーザー・マッピングの登録』
- 419 ページの『Teradata データ・ソースのユーザー・マッピングの作成』
- 83 ページの『BioRS データ・ソースのユーザー・マッピングの登録』
- 182 ページの『DB2 データ・ソースのユーザー・マッピングの作成』

- 267 ページの『Extended Search データ・ソースのユーザー・マッピングの登録』
- 318 ページの『Informix データ・ソースのユーザー・マッピングの作成』
- 335 ページの『Microsoft SQL Server データ・ソースのユーザー・マッピングの作成』
- 348 ページの『ODBC データ・ソースのユーザー・マッピングの作成』
- 363 ページの『OLE DB データ・ソースのユーザー・マッピングの作成』
- 375 ページの『Oracle データ・ソースのユーザー・マッピングの作成』
- 391 ページの『Sybase データ・ソースのユーザー・マッピングの作成』

データ・ソースのニックネームの登録

データ・ソースのニックネームの登録

ニックネームの登録作業は、一般に構成作業で最も複雑な作業です。ニックネームの登録に関するステップおよび要件は、データ・ソースごとに異なります。

推奨事項: ニックネームを登録するには、DB2 コントロール・センターを使用してください。ほとんどのデータ・ソースに関して、DB2 コントロール・センターの Discover ツールを使用できます。Discover ツールは、ニックネームの登録を必要とする可能性があるデータ・ソース・オブジェクトを素早く識別するのに役立ちます。

アクセスするデータ・ソース・オブジェクトごとに、ニックネームを登録しなければなりません。

データ・ソース・オブジェクトには、以下のように、リレーショナルのものと非リレーショナルのものがあります。

- リレーショナル・データ・ソース・オブジェクトとしては、たとえばデータベース表、ビュー、およびシノニム (Informix および Oracle のみ) があります。
- 非リレーショナル・データ・ソース・オブジェクトとしては、たとえば BLAST 可能なデータベース、Documentum Docbase 内のオブジェクトおよび登録済み表、Microsoft Excel ファイル (.xls)、表構造ファイル (.txt)、および XML タグ・ファイルがあります。

フェデレーテッド・データベースにある表とビューは、ローカル・オブジェクトです。これらのオブジェクトにはニックネームを登録しません。これらのオブジェクトの照会では実際のオブジェクト名を使用します。

データ・ソース・オブジェクト とは、次のものです。

- フェデレーテッド・サーバー上の別の DB2® データベース・インスタンス内の表およびビュー。
- 別のサーバー上の DB2 インスタンス内の表およびビュー。
- 別のデータ・ソース (例: Oracle、Sybase、Documentum、および ODBC) にあるデータ・ソース・オブジェクト。

これらのオブジェクトにはニックネームを登録しなければなりません。データ・ソース・オブジェクトは、リモート・オブジェクト と呼ばれることがあります。

フェデレーテッド・サーバーに対して分散要求をサブミットする場合、その要求はデータ・ソース・オブジェクトをそのニックネームで参照します。ニックネームはデータ・ソース側の特定のオブジェクト名にマップされています。このマッピングにより、ニックネームをデータ・ソース名で修飾する必要がなくなります。クライアント・アプリケーションまたはエンド・ユーザーは、データ・ソース・オブジェクトのロケーションを意識する必要はありません。ニックネームはデータ・ソース・オブジェクトの別名ではありません。ニックネームは、フェデレーテッド・サーバーがこれらのオブジェクトを参照するために使用するポインターです。

たとえば、ニックネーム *DEPT* が Informix データベース表の *NFX1.PERSON.DEPT* を表すように定義すると、`SELECT * FROM DEPT` というステートメントをフェデレーテッド・サーバーから使用できます。しかし、`SELECT * FROM NFX1.PERSON.DEPT` というステートメントは使用できません。

ニックネームを登録する場合、そのニックネームに関するメタデータ情報はフェデレーテッド・データベース・システム・カタログに保管されます。リレーショナル・データ・ソース・オブジェクトの場合、リモート・サーバーからカタログ・データが検索され、フェデレーテッド・データベース・システム・カタログにそのデータが保管されます。非リレーショナルのデータ・ソースの場合、データ・ソース情報のフェデレーテッド・データベース・システム・カタログへの保管のされ方はデータ・ソースによって異なります。情報をリモート・サーバーから検索したり、この情報を `CREATE NICKNAME` ステートメントに含めなければならないこともあります。

SQL コンパイラーはこのメタデータ情報を使用して、データ・ソース・オブジェクトにアクセスできるようにします。たとえば、索引を含む表でニックネームが登録されると、その索引に関連するメタデータ情報がフェデレーテッド・データベース・システム・カタログに保管されます。ニックネームを照会すると、SQL コンパイラーは索引メタデータ情報 (索引キー内の各列の名前など) を使用します。

手順:

ニックネームは、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- DB2 コントロール・センターからこのタスクを行うには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、または「ニックネーム」フォルダーを右マウス・ボタンでクリックし、「作成」をクリックします。Discover ツールを使用して、ニックネームを作成するオブジェクトを識別します。
- このタスクを DB2 コマンド行から行うには、`CREATE NICKNAME` ステートメントを使用します。

同じデータ・ソース・オブジェクトに、複数のニックネームを定義することができます。

関連タスク:

- 402 ページの『表構造ファイルのニックネームの登録』
- 197 ページの『Documentum データ・ソースのニックネームの登録』
- 247 ページの『Excel データ・ソースのニックネームの登録』

- 121 ページの『BLAST データ・ソースのニックネームの登録』
- 474 ページの『XML データ・ソースのニックネームの登録』
- 224 ページの『Entrez データ・ソースのニックネームの登録』
- 422 ページの『Teradata 表およびビューのニックネームの登録』
- 84 ページの『BioRS データ・ソースのニックネームの登録』
- 185 ページの『DB2 表およびビューのニックネームの登録』
- 268 ページの『Extended Search データ・ソースのニックネームの登録』
- 296 ページの『HMMER データ・ソースのニックネームの登録』
- 320 ページの『Informix 表、ビュー、およびシノニムのニックネームの登録』
- 338 ページの『Microsoft SQL Server 表およびビューのニックネームの登録』
- 351 ページの『ODBC データ・ソース表およびビューのニックネームの登録』
- 378 ページの『Oracle 表およびビューのニックネームの登録』
- 393 ページの『Sybase 表およびビューのニックネームの登録』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 25 ページの『フェデレーテッド・オブジェクトの命名規則』

ニックネームに対するニックネームの作成

場合によっては、ニックネームに対してニックネームを作成する必要がある場合があります。

手順:

たとえば、AIX[®] を使用するフェデレーテッド・サーバーと Windows を使用するフェデレーテッド・サーバーがあり、この両方のフェデレーテッド・サーバーから Excel のスプレッドシートにアクセスしたいとします。しかし、Excel ラッパーは Windows を使用するフェデレーテッド・サーバー上でしかサポートされません。AIX フェデレーテッド・サーバーから Excel スプレッドシートにアクセスするには、以下のステップに従ってください。

1. Windows フェデレーテッド・サーバー上で DB2 Information Integrator をインストールする。
2. Excel データ・ソースにアクセスするように Windows フェデレーテッド・サーバーを構成する。
3. Windows フェデレーテッド・サーバー上で Excel スプレッドシートのニックネームを作成する。
4. AIX フェデレーテッド・サーバー上で DB2 Information Integrator をインストールする。
5. DB2 ファミリー・データ・ソースにアクセスするように AIX フェデレーテッド・サーバーを構成する。
6. AIX フェデレーテッド・サーバー上で Windows フェデレーテッド・サーバー上の Excel ニックネームのニックネームを作成する。

非リレーショナル・データ・ソースのニックネーム列の指定

いくつかの非リレーショナル・データ・ソースでは、ニックネームを登録する際に列のリストを定義しなければなりません。指定する各列は、データ・ソース・オブジェクト内の特定のフィールド、列、またはエレメントにマップされます。

いくつかの非リレーショナル・データ・ソースで使用されるラッパーは、固定入出力列のセットを必要とします。ニックネームを登録して、フェデレーテッド・データベース・システム・カタログに追加される時に、固定列が自動的に定義されます。

手順:

ニックネームを登録する際に列のリストを定義するには、列名およびデータ・タイプを指定します。ニックネーム列にオプションを指定する場合があります。

関連タスク:

- 71 ページの『データ・ソースのニックネームの登録』

オプションの構成ステップ

データ・ソースへアクセスするためにフェデレーテッド・サーバーを構成する際に、いくつかのオプションのステップを行わなければならない場合があります。

索引の指定:

索引がないオブジェクトに索引の指定を定義します。たとえば、表に新しい索引ができた時に、または通常は索引を持たないデータ・ソース・オブジェクト (ビューなど) に索引の指定を作成します。

データ・タイプ・マッピング:

代替データ・タイプ・マッピングを指定できるのは、リレーショナル・データ・ソースだけです。

デフォルトのデータ・タイプ・マッピングを使用せずに、代替のデータ・タイプ・マッピングを指定します。特定のデータ・ソース・オブジェクト (データベース内の特定の表など) にのみ使用するマッピングを指定することができます。

関数マッピング:

関数マッピングを指定できるのは、リレーショナル・データ・ソースだけです。

デフォルトの関数マッピングを使用せずに、代替の関数マッピングを定義します。これは、強制的に DB2® にデータ・ソース側でユーザー定義関数を使用させたい場合に特に便利です。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・サーバーでのデータ・タイプ・マッピング』

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムでの関数マッピング』
- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムでの索引の指定』
- 61 ページの『データ・ソースの構成の近道』

第 6 章 BioRS データ・ソースへのアクセスの構成

この章では、BioRS データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。BioRS データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- BioRS の概要を説明します。
- 実行する必要のあるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- BioRS ラッパーに関連するエラー・メッセージをリストします。

BioRS とは何か

BioRS は、ドイツ Biomax 社が開発した照会および検索システムです。BioRS を使用すると、フラット・ファイルやリレーショナル・データベースを含む、複数のデータ・ソースから情報を検索できます。通常 SwissProt や GenBank のような公用データは、フラット・ファイルとして BioRS システムにダウンロードします。BioRS は公用データ・ソースと専用データ・ソース (組織独自で保守する専用データベースなど) を共通環境に統合できます。

BioRS システムに統合されたデータ・ソースは、データ・バンクと呼ばれます。各データ・バンクの項目に含まれるエレメントは、まとめてスキーマと呼ばれます。索引付けされているデータ・バンクのエレメントは、BIORS.CONTAINS、BIORS.CONTAINS_GE、および BIOR.S.CONTAINS_LE 関数で使用できます。BioRS 関数は SELECT ステートメントの WHERE 文節で指定されます。索引付けされていないエレメントは、SELECT リストおよび WHERE 文節内の他の述部で参照できます。索引付けされていないエレメントは、フェデレーテッド・サーバーによって処理されます。

データ・バンクの項目同士を関連付け、BioRS システムでデータ・バンクを結合することも可能です。

BioRS データ・バンクには、親子関係を指定できます (ネスト可能)。親子関係では、子データ・バンクは PARENT と呼ばれる Reference データ・タイプ・エレメントを持ちます。PARENT エレメントは親データ・バンクの _ID_ エレメントを参照します。ネスト化されたデータ・バンクとネスト化されていないデータ・バンクが収容するデータの違いは、この定義済み PARENT エレメントのみです。

BioRS は BioRS データ・バンクのデータを照会できる Web ベースのインターフェースを提供します。BioRS ラッパーは、BioRS の Web ベースのインターフェースと同じアプリケーション・プログラミング・インターフェース (API) を使用して情報を照会します。

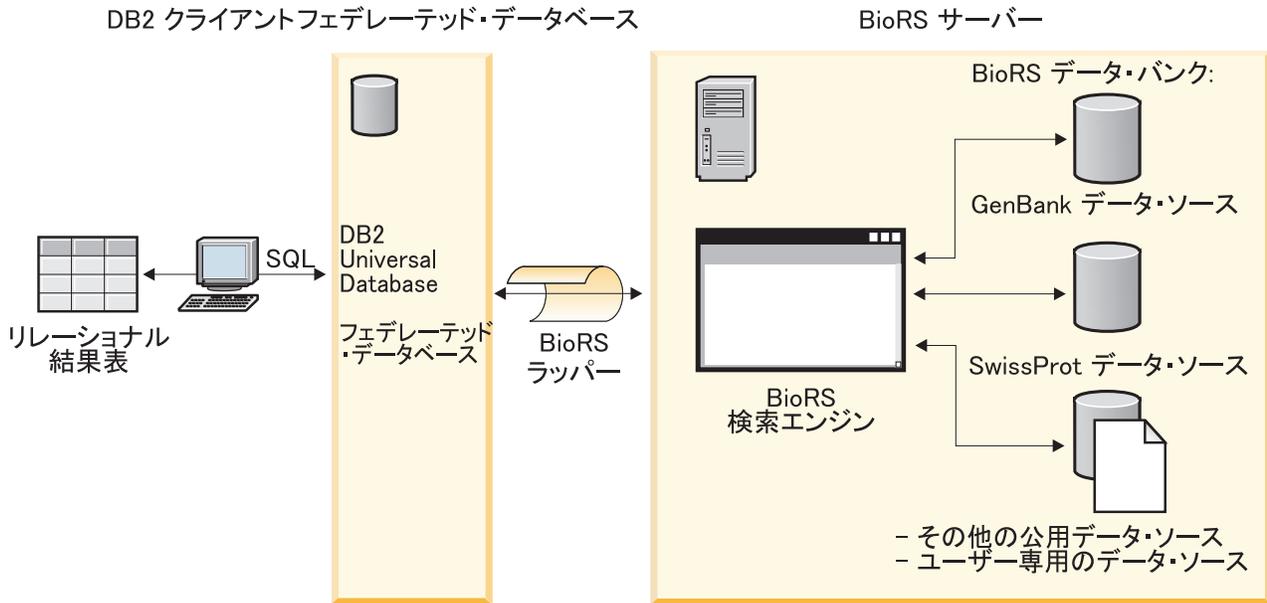


図 6. BioRS ラッパーが機能する構造

ユーザーまたはアプリケーションは、クライアントから SQL ステートメントを使用した照会をサブミットします。照会は BioRS ラッパーがインストールされたフェデレーテッド・システムに送信されます。照会の構成の仕方によっては、フェデレーテッド・サーバー と BioRS サーバーの両方が照会の処理に使用される場合があります。BioRS サーバーは、フェデレーテッド・システムとは異なるコンピュータ上にある場合もあります。ただし、各照会ごとに、フェデレーテッド・システムから BioRS サーバーに認証情報を提供する必要があります。認証情報は、ユーザー ID とパスワードの組み合わせか、非認証標識 (通常はゲスト・アカウント) のいずれかです。

BioRS ラッパーは BioRS バージョン 5.0.14 と連動します。

BioRS 製品の詳細については、Biomax の Web サイト <http://www.biomax.com> を参照してください。

関連タスク:

- 78 ページの『フェデレーテッド・サーバーへの BioRS データ・ソースの追加』

関連資料:

- 92 ページの『BioRS ラッパー - 照会例』

フェデレーテッド・サーバーへの BioRS の追加

フェデレーテッド・サーバーへの BioRS データ・ソースの追加

BioRS データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

DB2 コントロール・センターまたは DB2 コマンド行を使用して、BioRS データ・ソースへアクセスするようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

BioRS データ・ソースをフェデレーテッド・サーバーに追加するには、次のようにします。

1. BioRS ラッパーのカスタム関数を登録します。
2. BioRS ラッパーを登録します。
3. BioRS サーバー定義を登録します。
4. オプション: ユーザー・マッピングを作成します。
5. BioRS データ・バンクのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 79 ページの『BioRS ラッパーのカスタム関数の登録』
- 81 ページの『BioRS ラッパーの登録』
- 82 ページの『BioRS データ・ソースのサーバー定義の登録』
- 83 ページの『BioRS データ・ソースのユーザー・マッピングの登録』
- 84 ページの『BioRS データ・ソースのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

BioRS ラッパーのカスタム関数の登録

BioRS ラッパーのカスタム関数の登録は、BioRS データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

BioRS ラッパーのカスタム関数は以下のとおりです。

- `biors.contains`

- `biors.contains_le`
- `biors.contains_ge`
- `biors.search_term`

前提条件:

- BioRS ラッパーのカスタム関数はすべて、`biors` というスキーマ名を使用して登録する必要があります。
- BioRS ラッパーがインストールされている各フェデレーテッド・データベースに対して、各カスタム関数を 1 回ずつ登録する必要があります。

手順:

カスタム関数を登録するには、キーワード `AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION` を使用して `CREATE FUNCTION` ステートメントを発行してください。

各関数の完全修飾名は、`biors.function_name` です。

以下の例では、`CONTAINS` 関数の 1 つのバージョンを登録します。

```
CREATE FUNCTION biors.contains (varchar(), varchar())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

カスタム関数を登録するには、サンプル・ファイル `create_function_mappings.ddl` を使用します。このサンプル・ファイルは `sqllib/samples/lifesci/biors` ディレクトリー内にあります。サンプル・ファイルには、カスタム関数ごとの定義が含まれています。この DDL ファイルを実行して、BioRS ラッパーがインストールされている各フェデレーテッド・データベースにカスタム関数を登録できます。

この一連の作業における次のタスクは、『BioRS ラッパーの登録』です。

関連タスク:

- 81 ページの『BioRS ラッパーの登録』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『`CREATE FUNCTION` (ソースまたはテンプレート) ステートメント』
- 87 ページの『カスタム関数と BioRS 照会』
- 92 ページの『BioRS ラッパー - 照会例』
- 80 ページの『カスタム関数表 - BioRS ラッパー』

カスタム関数表 - BioRS ラッパー

BioRS カスタム関数を登録するには、`CREATE FUNCTION` ステートメントを使用します。

以下の表では、4 つの BioRS カスタム関数と、関数を登録する際に指定できるデータ・タイプの例がリストされています。

関数に指定できる最初のデータ・タイプは、索引付けされた列です。関数に指定できる 2 番目のデータ・タイプは、検索語です。

表 12. BioRS ラッパーのカスタム関数

関数	説明
biors.contains (varchar(), varchar())	索引付けされている列を検索して、指定する値と等しい (BioRS 照会セマンティクスに従って) 値を探します。最初の引き数は索引付けされている列への参照でなければならず、2 番目の引き数は指定する値になります。
biors.contains (varchar(), char())	
biors.contains (varchar(), date)	
biors.contains (varchar(), timestamp)	
biors.contains_LE (varchar(), varchar())	索引付けされている列を検索して、指定する値以下の (BioRS 照会セマンティクスに従って) 値を探します。最初の引き数は索引付けされている列への参照でなければならず、2 番目の引き数は指定する値になります。
biors.contains_LE (varchar(), smallint)	
biors.contains_LE (varchar(), bigint)	
biors.contains_LE (varchar(), decimal)	
biors.contains_LE (varchar(), double)	
biors.contains_LE (varchar(), real)	
biors.contains_GE (char(), char())	索引付けされている列を検索して、指定する値以上の (BioRS 照会セマンティクスに従って) 値を探します。最初の引き数は索引付けされている列への参照でなければならず、2 番目の引き数は指定する値になります。
biors.contains_GE (char(), date)	
biors.contains_GE (char(), timestamp)	
biors.contains_GE (char(), integer)	
biors.contains_GE (char(), smallint)	
biors.contains_GE (clob(), date)	
biors.search_term (varchar(), varchar())	
biors.search_term (varchar(), char())	
biors.search_term (char(), varchar())	
biors.search_term (char(), char())	

関連タスク:

- 79 ページの『BioRS ラッパーのカスタム関数の登録』

BioRS ラッパーの登録

BioRS ラッパーの登録は、BioRS データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

BioRS データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバーで、biors_wrapper という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER biors_wrapper LIBRARY 'libdb2lsbiors.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『BioRS ラッパーのライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『BioRS ラッパーのサーバー定義の登録』です。

関連資料:

- 82 ページの『BioRS ラッパーのライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

BioRS ラッパーのライブラリー・ファイル

次の表では、BioRS ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、libdb2lsbiors.a、libdb2lsbiorsF.a、および libdb2lsbiorsU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 13. BioRS ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパーのライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lsbiors.a
Linux	/opt/IBM/db2/V8.1/lib	libdb2lsbiors.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lsbiors.so
Windows	%DB2PATH%\bin	db2lsbiors.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連資料:

- 45 ページの『ラッパー・ライブラリー・ファイル』

BioRS データ・ソースのサーバー定義の登録

BioRS データ・ソースのサーバー定義の登録は、BioRS をフェデレーテッド・システムに追加する作業の一部です。ラッパーの登録後、対応するサーバー定義を登録する必要があります。

手順:

フェデレーテッド・データベースで BioRS サーバー定義を登録するには、CREATE SERVER ステートメントを発行します。

例えば、以下のようになります。

```
CREATE SERVER brs_server WRAPPER wrap_biors OPTIONS(NODE 'biors_server2.com');
```

この一連の作業における次のタスクは、『BioRS データ・ソースのユーザー・マッピングの登録』です。

関連タスク:

- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 587 ページの『CREATE SERVER ステートメント・オプション - BioRS ラッパー』

BioRS データ・ソースのユーザー・マッピングの登録

ユーザー・マッピングの登録は、BioRS をフェデレーテッド・システムに追加する作業の一部です。

アカウントへのアクセス方式または BioRS システムで使用している方式により、ユーザー・マッピングを作成する必要がない場合があります。

- BioRS サーバーですべてのユーザー・アカウントにゲスト・アクセスを設定している場合、DB2 Information Integrator にユーザー・マッピングを作成する必要はありません。
- BioRS サーバーでユーザー・アカウントの認証に ID とパスワードを使用している場合、BioRS ラッパーを使用するアカウントのユーザー・マッピングをフェデレーテッド・データベースに作成する必要があります。
- BioRS サーバーでゲストおよび認証ユーザー・アカウントの両方を使用している場合、BioRS ラッパーを使用するアカウントに対して、認証ユーザー・アカウントのユーザー・マッピングをフェデレーテッド・データベースに作成する必要があります。

ユーザー・マッピングは、BioRS ラッパーを使用して BioRS データ・ソースを照会するユーザーまたはアプリケーションのアクセスを認証します。ユーザーまたはアプリケーションが SQL 照会を登録済み BioRS ニックネームにサブミットし、そのユーザーまたはアプリケーションに対するユーザー・マッピングが未定義の場合、BioRS ラッパーはデフォルトのユーザー ID とパスワードを使用して、リモート BioRS サーバーからデータを取り出そうとします。照会対象のデータ・バンクで認証が必要な場合、エラー・メッセージが戻される可能性があります。

正しいユーザー ID とパスワードを確実に BioRS サーバーに渡すには、BioRS データ・ソースの検索を許可されたユーザーのユーザー・マッピングをフェデレーテッド・システムに作成します。ユーザー・マッピングを作成する際、パスワードは暗号化された形式でフェデレーテッド・データベースのシステム・カタログ表に格納されます。

手順:

BioRS ユーザー・マッピングを登録するには、CREATE USER MAPPING ステートメントを使用します。

たとえば、次に示す CREATE USER MAPPING ステートメントは、Biors_Server1 サーバーでユーザー Charlie をユーザー Charlene にマップします。

```
CREATE USER MAPPING FOR Charlie SERVER Biors_Server1
OPTIONS(REMOTE_AUTHID 'Charlene', REMOTE_PASSWORD 'Charlene_pw');
```

独自のユーザー・マッピングを定義することもできます。以下の例の USER は、USER というユーザー名ではなく、現行のユーザーを示すキーワードです。

```
CREATE USER MAPPING FOR USER SERVER Biors_Server1
OPTIONS(REMOTE_AUTHID 'Yudong', REMOTE_PASSWORD 'Yudong_pw')
```

この一連の作業における次のタスクは、『BioRS データ・ソースのニックネームの登録』です。

関連タスク:

- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』
- 588 ページの『CREATE USER MAPPING ステートメント・オプション - BioRS ラッパー』

BioRS データ・ソースのニックネームの登録

BioRS データ・ソースのニックネームの登録は、BioRS をフェデレーテッド・サーバーに追加する作業の一部です。

サーバー定義の登録後、アクセスする各データ・ソースに対応するニックネームを登録する必要があります。ニックネームは、照会で BioRS データ・ソースを参照する場合に使用します。

BioRS システムに統合されたデータ・ソースは、BioRS 内でデータ・バンクとして参照されます。BioRS 内のデータ・バンクはフェデレーテッド・システム上のニックネームと同じです。

前提条件:

- CREATE NICKNAME ステートメントが必要とする構文に、BioRS データ・バンク名が準拠していない場合、ニックネームを登録する際に REMOTE_OBJECT ニックネーム・オプションを使用しなければなりません。
- CREATE NICKNAME ステートメントが必要とする構文に、BioRS エlement名が準拠していない場合、ニックネームを登録する際に ELEMENT_NAME 列オプションを使用しなければなりません。

制約事項:

BioRS AllText エlementをニックネームの1列目に使用しないでください。その他の列(2列目や3列目)では使用できます。

手順:

BioRS ニックネームを登録するには、CREATE NICKNAME ステートメントを使用します。

BioRS ニックネームを作成する場合、ニックネーム列のリストを定義してください。指定したニックネーム列は、特定の BioRS データ・バンク形式のエレメントに対応する必要があります。BioRS では、エレメントに対して、使用可能な 5 つのデータ・タイプ (Text、Number、Date、Author、および Reference) を定義しています。BioRS データ・タイプは DB2 データ・タイプ CHAR、CLOB、または VARCHAR にのみマップできます。

BioRS データ・バンクのニックネームを登録する最も簡単な方法は、BioRS データ・バンクと同じ名前のニックネームを付けることです。例えば、以下のようになります。

```
CREATE NICKNAME SwissProt
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
FOR SERVER brs_server;
```

BioRS データ・バンクである SwissProt がニックネームの名前です。

この単純な CREATE NICKNAME 構文を使用した場合、1 つの DB2 スキーマにつき 1 つのニックネーム・ファミリーのみを使用できます。たとえば、親子関係になっている 2 つのデータ・バンクがあるとします。データ・バンクは SWISSPROT と SPFEAT です。これらのデータ・バンクはファミリーを形成します。CREATE NICKNAME ステートメントのデフォルトの構文を使用する場合、SWISSPROT データ・バンクに 1 つのニックネーム (SWISSPROT) および SPFEAT データ・バンクに 1 つのニックネーム (SPFEAT) を持つようになります。スキーマ内で SWISSPROT に複数のニックネームを持つには、REMOTE_OBJECT オプションを使用しなければなりません。

REMOTE_OBJECT ニックネーム・オプションでは、ニックネームに関連する BioRS データ・バンク名を指定します。REMOTE_OBJECT オプションで指定する名前によって、ニックネームのスキーマと BioRS データ・バンクが決まります。REMOTE_OBJECT オプションはニックネーム間の関係も指定します。

以下の例は、前の例と同じニックネーム特性の設定を使用していますが、ニックネームを変え、REMOTE_OBJECT オプションを使用してニックネームを定義する BioRS データ・バンクを指定しています。

```
CREATE NICKNAME NewSP
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR(128),
  ENTRYDATE VARCHAR (64))
FOR SERVER brs_server
  OPTIONS (REMOTE_OBJECT 'SwissProt');
```

ニックネームを作成する BioRS データ・バンクごとに、このステップを繰り返します。

一連の作業におけるステップは、これで終わりです。

関連概念:

- 98 ページの『BioRS 統計情報』

関連タスク:

- 99 ページの『BioRS ニックネーム・カーディナリティー統計の更新』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 91 ページの『BioRS AllText エlement』
- 86 ページの『CREATE NICKNAME ステートメント - BioRS ラッパーの例』
- 589 ページの『CREATE NICKNAME ステートメント構文 - BioRS ラッパー』

CREATE NICKNAME ステートメント - BioRS ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して BioRS ラッパーのニックネームを登録する方法を例示します。

例 1

以下の例は、DB2 Information Integrator 構文の規格に適合しないリモート BioRS データ・バンクのニックネームを作成する方法を示します。

```
CREATE NICKNAME SwissFT
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (128),
  ENTRYDATE VARCHAR (64),
  FtLength VARCHAR (16),
  FOR SERVER biors1
  OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

ニックネームの名前は SwissFT です。表の列は ID、ALLTEXT、ENTRYDATE、および FtLength です。ELEMENT_NAME 列オプションは ID 列に対して指定します。BioRS エlementが、列名に対する有効な DB2 フェデレーテッド構文の規格に適合しない場合、ELEMENT_NAME オプションを指定する必要があります。この例では、BioRS エlement `_ID_` は DB2 フェデレーテッド構文に準拠しますが、`_ID_` は DB2 Information Integrator ユーザーが混乱する可能性のある名前です。ID は簡潔でわかりやすい名前です。通常、以下のような場合に ELEMENT_NAME オプションを使用します。

- BioRS エlement名が有効な DB2 フェデレーテッド構文の規格に適合しない場合
- BioRS エlement名の大文字小文字の区別が、設定した DB2 フェデレーテッド・システムの規格に適合しない場合
- BioRS エlement名が DB2 Information Integrator ユーザーにとって不明な場合

さらに、REMOTE_OBJECT オプションは、ニックネームが同じ BioRS データ・バンクの名前の指定にも利用します。BioRS データ・バンク名が、有効な DB2 フェデレーテッド構文の規格に適合しない場合、REMOTE_OBJECT オプションを指定する必要があります。この例では、データ・バンク名「SwissProt.Features」は有効な DB2 フェデレーテッド構文の規格に合いません。通常、以下のような場合に REMOTE_OBJECT オプションを使用します。

- BioRS データ・バンク名の大文字小文字の区別が、設定した DB2 フェデレーテッド・システムの規格に適合しない場合

- BioRS データ・バンク名が有効な DB2 フェデレーテッド構文の規格に適合しない場合
- BioRS データ・バンク名が DB2 Information Integrator ユーザーにとって不明白な場合

例 2

以下の例は、他の BioRS データ・バンクにリンクする BioRS データ・バンクを使用する表のニックネームを作成する方法を示します。

```
CREATE NICKNAME SwissFT2
  (ID VARCHAR(32) OPTIONS (ELEMENT_NAME '_ID_'),
  ALLTEXT VARCHAR (1200),
  FtKey VARCHAR (32),
  FtLength VARCHAR (64),
  FtDescription VARCHAR (128),
  Parent VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt'))
FOR SERVER biors1
OPTIONS (REMOTE_OBJECT 'SwissProt.Features');
```

ニックネームの名前は SwissFT2 です。表列は ID、 ALLTEXT、 FtKey、 FtLength、 FtDescription、 および Parent です。ELEMENT_NAME 列オプションは ID 列に対して指定します。REMOTE_OBJECT オプションは、ニックネームが対応する BioRS データ・バンクの名前の指定にも利用します。

さらに、Parent 列は REFERENCED_OBJECT オプションを使用します。このオプションは BioRS Reference データ・タイプ・エレメントに対応する列に指定する必要があります。REFERENCED_OBJECT オプションは、列が参照する BioRS データ・バンクの名前を指定します。この場合、Parent エレメントは BioRS SwissProt データ・バンクを参照します。

関連タスク:

- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 589 ページの『CREATE NICKNAME ステートメント構文 - BioRS ラッパー』

BioRS データ・ソースの照会およびカスタム関数

カスタム関数と BioRS 照会

フェデレーテッド環境は 2 つの照会エンジンを使用します。BioRS ラッパーでの照会エンジンは DB2 Universal Database と BioRS です。以下に挙げる 4 つの BioRS カスタム関数を使用して、述部を BioRS エンジンにプッシュ・ダウンすることを指定できます。

- BIOR.S.CONTAINS
- BIOR.S.CONTAINS_LE
- BIOR.S.CONTAINS_GE
- BIOR.S.SEARCH_TERM

4 つのカスタム関数は BioRS スキーマに登録されます。BioRS スキーマを使用して関数を参照してください。

カスタム関数 `BIORS.CONTAINS`、`BIORS.CONTAINS_LE`、および `BIORS.CONTAINS_GE` には検索語列引き数と照会テキスト引き数が必要です。以下に、`BIORS.CONTAINS` ステートメントの例を示します。

```
BIORS.CONTAINS (<search term column>,<query term>)
```

検索語列引き数値は、索引付けされた BioRS 列を参照します。索引付けされていない列を使用すると、エラー・メッセージ `SQL30090N` (「操作がアプリケーション実行環境で無効です。」) が表示されます。

照会語引き数の値は、リテラル、ホスト変数、または列参照のいずれかになります。演算またはストリング連結は使用できません。また、使用する検索語列が `NULL` 値を許可するよう定義済みでも、照会語引き数値は `NULL` を取れません。

照会語引き数では、大文字小文字は区別されません。

照会語引き数で有効なデータ・タイプと形式は、使用される検索語の BioRS データ・タイプによって決まります。BioRS では、使用可能な 5 つのデータ・タイプ (`Text`、`Author`、`Date`、`Number`、および `Reference`) を定義しています。BioRS データ・タイプおよびデータ・タイプごとの有効な関数照会語を、表 14 にリストします。

表 14. BioRS データ・タイプと有効なカスタム関数照会語

検索語列のデータ・タイプ	有効な照会語	形式
<code>Text</code>	<code>VARCHAR()</code> または <code>CHAR()</code>	BioRS テキスト用語 (ワイルドカードを含む)。
<code>Author</code>	<code>VARCHAR()</code> または <code>CHAR()</code>	"<last>、 <init>" 書式の BioRS 作成者参照。 "<last>" は作成者の姓です。 "<init>" は作成者のイニシャルですが、ピリオドはつけません。コンマとイニシャルの間には空白文字を挿入できます。 また、コンマやイニシャルを含めずに <last> だけを指定することもできます。
<code>Date</code>	<code>VARCHAR()</code> 、 <code>CHAR()</code> 、 <code>DATE</code> 、または <code>TIMESTAMP</code>	文字ストリングでは、DB2 形式の日付、 <code>yyyy/mm/dd</code> 。
<code>Number</code>	<code>VARCHAR()</code> または <code>CHAR()</code> 、 <code>INTEGER</code> 、 <code>SMALLINT</code> 、 <code>BIGINT</code> 、 <code>REAL</code> 、 <code>DOUBLE</code> 、 <code>DECIMAL</code>	DB2 形式の数字。
<code>Reference</code>	<code>VARCHAR()</code> または <code>CHAR()</code>	BioRS テキスト用語。

BioRS データ・タイプ検索語列と照会語引き数などのその他の組み合わせを使用すると、エラー・メッセージ `SQL30090N` (「操作がアプリケーション実行環境で無効です。」) が表示されます。表 14 に示した組み合わせでのみ使用できます。

Text、Author、および Reference データ・タイプ検索語列の照会語引き数は、BioRS 照会言語パターンと一致する必要があります。BioRS では、英数字とワイルドカードで照会語引き数を作成できます。BIORS.CONTAINS は、疑問符 (?) とアスタリスク (*) の 2 つのワイルドカードをサポートします。

? ワイルドカードは英数字 1 文字に相当します。例えば、述部 BioRS.CONTAINS (description, 'bacteri?')=1 は用語 bacteria (バクテリア) と一致しますが、bacterial (バクテリアの) とは一致しません。

* ワイルドカードは、ゼロまたは複数の文字に相当します。例えば、述部 BioRS.CONTAINS (description, 'bacteri*')=1 は用語 bacteri (バクテリ)、bacteria (バクテリア)、および bacterial (バクテリアの) と一致します。

BioRS 照会言語パターンの詳細については、ご使用の BioRS の資料を参照してください。

BIORS.CONTAINS 関数はすべての BioRS 列型に指定できます。

BIORS.CONTAINS_GE と BIOR.CONTAINS_LE カスタム関数は基本 BioRS データ・タイプが Number もしくは Date である列にのみ指定できます。BIORS.CONTAINS_GE 関数は、照会語引き数が表す値より大きいか等しい値が入っている列の行を選択します。BIORS.CONTAINS_LE 関数は、照会語引き数が表す値より小さいか等しい値が入っている列の行を選択します。

BIORS.CONTAINS、BIORS.CONTAINS_GE、および BIOR.CONTAINS_LE 関数は整数の結果を戻します。3 つの CONTAINS 関数のいずれかを述部で使用する場合、戻り値は = または <> 演算子を使用して値 1 と比較されます。例えば、以下のようになります。

```
SELECT * FROM s.MySP WHERE BIOR.CONTAINS (s.AllText, 'mucus') = 1;
```

表現形式 NOT (BioRS.Contains (col,value) = 1) は、BioRS.CONTAINS (col,value) <> 1 と同じです。

BIORS.SEARCH_TERM 関数を実行すると、別の方法では実行が難しい照会を実行できます。BioRS 形式を使用した検索語の指定にこの関数を使用できます。

BIORS.SEARCH_TERM 関数には 2 つの引き数が必要です。最初の引き数は、用語が適用されるニックネームの _ID_ 列への参照です。2 番目の引き数は、データ・バンク名を持たない用語を含む文字ストリングです。

以下の例では、SeqLength エlementが 100 以上の値を含む MyEMBL データ・バンクの項目に対するすべての列を選択しています。

```
SELECT * FROM MyEMBL s WHERE  
  BIOR.SEARCH_TERM (s.ID, '[SeqLength GREATER number:100;]') = 1;
```

以下の例では、Swiss ニックネームから MolWeight エlementの値が 100368 以上の MolWeight 列を選択しています。

```
SELECT s.molweight FROM Swiss s WHERE  
  BIOR.SEARCH_TERM (s.ID, '[MolWeight GREATER number:100368;]') = 1;
```

関連概念:

- 「フェデレーテッド・システム・ガイド」の『プッシュダウン分析』

- 97 ページの『BioRS ラッパー・パフォーマンスの最適化方法』
- 90 ページの『BioRS ラッパーの等価結合述部』

関連タスク:

- 79 ページの『BioRS ラッパーのカスタム関数の登録』

関連資料:

- 92 ページの『BioRS ラッパー - 照会例』
- 80 ページの『カスタム関数表 - BioRS ラッパー』

BioRS ラッパーの等価結合述部

BioRS カスタム関数を使用する場合、BioRS エンジンに対する述部を指定しなければなりません。ただし、例外が 1 つあります。それは、照会中に等価結合操作を行う場合です。結合 操作では、一致する列の値を基に複数の表からデータを取り出します。等価結合 は、式 = 式形式の結合条件をとる結合操作です。BioRS 照会では、等価結合条件は片方のデータ・バンクの `_ID_` エレメントともう一方のデータ・バンクの `Reference` タイプ・エレメントを含む必要があります。

例:

ここでは、ニックネーム定義のサンプルとサンプルのニックネームを使用した等価結合照会を例示します。

2 つの BioRS データ・バンク、`SwissProt` と `SwissProt.features` を照会します。`SwissProt.features` データ・バンクは `SwissProt` データ・バンクの子で、`Parent` と呼ばれるエレメントを含みます。`Parent` エレメントは `SwissProt` の `_ID_` エレメントが示す項目への参照を含みます。2 つのデータ・バンクに対して 2 つのニックネーム定義を登録します。

ニックネーム定義 1 は以下のとおりです。

```
CREATE NICKNAME tc600sprot (
  ID          VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
  AllText     VARCHAR (128),
  EntryDate   VARCHAR (128),
  Update      VARCHAR (128),
  Description  VARCHAR (1200),
  Crossreference VARCHAR (32),
  Authors     VARCHAR (256),
  Journal     VARCHAR (256),
  JournalIssue VARCHAR (64) OPTIONS (IS_INDEXED 'N'),
  PublicationYear VARCHAR (1024),
  Gene        VARCHAR (20) OPTIONS (IS_INDEXED 'Y'),
  Remarks     VARCHAR (1200),
  RemarkType  CHAR (20),
  CatalyticActivity VARCHAR (20),
  CoFactor    VARCHAR (64),
  Disease     VARCHAR (128),
  Function    VARCHAR (128),
  Pathway     VARCHAR (128),
  Similarity  VARCHAR (128),
  Complex     VARCHAR (64),
  FtKey       VARCHAR (32),
  FtDescription VARCHAR (128),
  FtLength    VARCHAR (256),
  MolWeight   VARCHAR (64),
  ProteinLen  VARCHAR (32) OPTIONS (ELEMENT_NAME 'Protein_length'),
```

```

|         Sequence          CLOB,
|         AccNumber        VARCHAR (32),
|         Taxonomy         VARCHAR (128),
|         Organelle        VARCHAR (128),
|         Organism         VARCHAR (128),
|         Keywords         VARCHAR (1200),
|         Localization     VARCHAR (128),
|         FtKey_count      VARCHAR (32)) FOR SERVER biors_server_600
|         OPTIONS (REMOTE_OBJECT 'SwissProt');

```

ニックネーム定義 2 は以下のとおりです。

```

| CREATE NICKNAME tc600feat (
|     ID      VARCHAR (32) OPTIONS (ELEMENT_NAME '_ID_'),
|     AllText VARCHAR (1200),
|     FtKey   VARCHAR (32),
|     FtLength VARCHAR (64),
|     FtDescription VARCHAR (128),
|     Parent  VARCHAR (32) OPTIONS (REFERENCED_OBJECT 'SwissProt')
|     FOR SERVER biors_server_600 OPTIONS (REMOTE_OBJECT 'SwissProt.features');

```

以下の照会は、等価結合にて両方のニックネームを参照します。

```

| SELECT s.ID, f.ID, f.FtKey FROM tc600sprot s, tc600feat f
|     WHERE BioRS.CONTAINS (s.AllText, 'anopheles') = 1
|           AND BioRS.CONTAINS (s.PublicationYear, 1997) = 1
|           AND BioRS.CONTAINS (f.FtKey, 'signal') = 1
|           AND f.Parent = s.ID;

```

前述した照会では、2 つの述部が tc600sprot ニックネーム (SwissProt データ・バンク) に適用されます。2 つの述部は用語 anopheles を含み、発行年が 1997 年である行をフィルターに掛けます。1 つの述部が tc600feat ニックネーム (SwissProt.features データ・バンク) に適用され、用語 signal を含む FtKey エレメントのある行をフィルターに掛けます。2 つのニックネームは条件 f.Parent = s.ID. を使用して結合されます。

最終結果セットには、これらの基準に適合し、フィーチャー項目が SwissProt データ・バンク内の一致する項目を参照する行のみが含まれます。

関連概念:

- 97 ページの『BioRS ラッパー・パフォーマンスの最適化方法』

関連資料:

- 87 ページの『カスタム関数と BioRS 照会』
- 92 ページの『BioRS ラッパー - 照会例』

BioRS AllText エレメント

BioRS システムのデータ・バンクにはそれぞれ、AllText と呼ばれるエレメントがあります。AllText エレメントは、BioRS がすべてのデータ・バンクに自動的に作成する、索引付けされたエレメントです。

AllText エレメントを使用することにより、特定の索引付けされたエレメントのみでなく、項目のテキストすべてを検索できます。例えば、用語 muscus を検索すると、単語 muscus が表題、要約、説明、または有機体に使用されている項目を返します。

AllText エlementを DB2 Information Integrator 照会で使用するには、AllText エlementをニックネーム列にマップする必要があります。CREATE NICKNAME ステートメントで列を指定する場合、AllText エlementをニックネーム列にマップします。AllText エlementへマップされているニックネーム列は、SELECT ステートメントで NULL 値を戻します。AllText エlementとして列を指定する場合、その列を CREATE NICKNAME ステートメントで宣言される最初の列にはなりません。

AllText エlementをニックネーム列に適切にマップすると、そのニックネーム列を CONTAINS カスタム関数呼び出しで使用できます。

関連タスク:

- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 92 ページの『BioRS ラッパー - 照会例』

BioRS ラッパー - 照会例

このトピックでは、ニックネーム `swiss` と `swissft` を使用した照会のサンプルをいくつか紹介します。

ニックネーム `swiss` は、次の CREATE NICKNAME ステートメントを使用して登録されています。

```
CREATE NICKNAME swiss
(
  ID                CHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  EntryDate         VARCHAR (15),
  Update            CLOB (15),
  Description        CLOB (15),
  Crossreference     CLOB (15),
  Authors            CLOB (15),
  Journal            VARCHAR (15),
  JournalIssue      VARCHAR (15),
  PublicationYear    CLOB (15),
  PublicationTitle   CLOB (15),
  Gene              CLOB (15),
  Remarks            CLOB (15),
  RemarkType        VARCHAR (15),
  CatalyticActivity VARCHAR (15),
  CoFactor           VARCHAR (15),
  Disease            VARCHAR (15),
  Function           CLOB (15),
  Pathway            VARCHAR (15),
  Similarity         CLOB (15),
  Complex            VARCHAR (15),
  FtKey              VARCHAR (15),
  FtDescription      CLOB (15),
  FtLength           VARCHAR (15),
  MolWeight          CHAR (15),
  Protein_Length    VARCHAR (15),
  Sequence           CLOB (15),
  AccNumber          VARCHAR (15),
  Taxonomy           CLOB (15),
  Organelle          VARCHAR (15),
  Organism           VARCHAR (15),
  Keywords           VARCHAR (15),
  Localization       VARCHAR (15),
  FtKey_count        VARCHAR (15),
```

```

AllText          CLOB (15)
)
FOR SERVER biors_server
  OPTIONS (REMOTE_OBJECT 'swissprot');

```

ニックネーム `swissft` は、次の `CREATE NICKNAME` ステートメントを使用して登録されています。

```

CREATE NICKNAME swissft
(
  ID          VARCHAR (30) OPTIONS (ELEMENT_NAME '_ID_'),
  FtKey       VARCHAR (15),
  FtLength    VARCHAR (15),
  FtDescription VARCHAR (15),
  Parent      VARCHAR (30) OPTIONS (REFERENCED_OBJECT 'swissprot'),
  AllText     CLOB (15)
)
FOR SERVER biors_server
  OPTIONS (REMOTE_OBJECT 'swissprot.features');

```

表 15 の照会と結果は、照会を構造化し、フェデレーテッド・システムと BioRS サーバー間でワークロードを最適化する方法を示します。

表 15. 異なる照会で同じ結果が得られる例

照会	結果
<pre>select s.id from Swiss s where biors.CONTAINS(s.id, '100K_RAT') = 1 fetch first 3 rows only</pre>	<pre>ID ----- 100K_RAT 1 record(s) selected.</pre>
<pre>select s.id from Swiss s where s.id LIKE '%100K_RAT%' fetch first 3 rows only</pre>	<pre>ID ----- 100K_RAT 1 record(s) selected.</pre>

表 15 のいずれの照会も同じ結果となります。ただし、1 番目の照会は、2 番目の照会と比べかなり短い時間で実行されます。最初の照会は入力述部の指定に `BIORS.CONTAINS` 関数を使用しています。その結果、BioRS は `swissprot` データ・バンクのデータを検索し、選択したデータを DB2 Information Integrator に渡します。2 番目の照会では、`LIKE` 入力述部が `Swiss` ニックネームに直接指定されています。そのため、BioRS は `swissprot` データ・バンク全体を DB2 Information Integrator に転送します。データ・バンク・コンテンツの転送後、DB2 Information Integrator がデータを選択します。

94 ページの表 16 に、`BIORS.CONTAINS` 関数にワイルドカード文字を使用した照会と結果の例を示します。異なるワイルドカード文字を使用していますが、94 ページの表 16 の照会結果はすべて同じになります。

表 16. *BIORS.CONTAINS* 関数にワイルドカードを使用した照会の例

照会	結果
<pre>select s.crossreference from Swiss s where biors.CONTAINS(s.crossreference, 'MEDLINE') = 1 fetch first 3 rows only</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected.</pre>
<pre>select s.crossreference from Swiss s where biors.CONTAINS(s.crossreference, '?ED?IN?') = 1 fetch first 3 rows only</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected.</pre>
<pre>select s.crossreference from Swiss s where biors.CONTAINS(s.crossreference, '*D*N*') = 1 fetch first 3 rows only</pre>	<pre>CROSSREFERENCE ----- NCBI_TaxID=1011 NCBI_TaxID=5875 NCBI_TaxID=4081 3 record(s) selected</pre>

表 17 に、*BIORS.CONTAINS* 関数を使用して BioRS Author データ・タイプ・エレメントの情報にアクセスする方法を示します。

表 17 のすべての照会構文はほとんど同じです。照会語の先頭のイニシャルの有無、および、ファーストネームと最後のイニシャルの間のスペースの数のみが異なります。

表 17. *BioRS Author* データ・タイプ列にアクセスする照会の例

照会	結果
<pre>select s.authors from Swiss s where biors.CONTAINS(s.authors, 'Mueller') = 1 fetch first 3 rows only</pre>	<pre>AUTHORS ----- Mueller D. Rehb Mayer K.F.X. Sc Zemmour J. Litt 3 record(s) selected.</pre>
<pre>select s.authors from Swiss s where biors.CONTAINS(s.authors, 'Mueller,D') = 1 fetch first 3 rows only</pre>	<pre>AUTHORS ----- 0 record(s) selected.</pre>
<pre>select s.authors from Swiss s where biors.CONTAINS(s.authors, 'Mueller ,D') = 1 fetch first 3 rows only</pre>	<pre>AUTHORS ----- 0 record(s) selected.</pre>
<pre>select s.authors from Swiss s where biors.CONTAINS(s.authors, 'Mueller, D') = 1 fetch first 3 rows only</pre>	<pre>AUTHORS ----- Mueller D. Rehb Zou P.J. Borovo Davies J.D. Mue 3 record(s) selected.</pre>

表 18 の照会と結果は、BIORS.CONTAINS 関数を使用して BioRS Date タイプ・エレメントの情報にアクセスする方法を示します。

BioRS Date タイプのフィールドがデータのシーケンスを含む場合、表 18 の 2 例目に見られるように、結果に余分な情報が含まれます。BioRS Numeric データ・タイプ・エレメント (Date と Number) には、複数の値を含めることができます。そのため、BioRS Date または Number エレメントに対して実行した照会結果も複数の値を含みます。値は常にスペースで区切ります。

表 18. BioRS Date データ・タイプ列にアクセスする照会の例

照会	結果
select e.entrydate from embl e where biors.CONTAINS(e.entrydate, date('11/01/1997')) = 1 fetch first 3 rows only	ENTRYDATE ----- 01-NOV-1997 01-NOV-1997 01-NOV-1997 3 record(s) selected.
select g.update from gen g where biors.CONTAINS(g.update, date('11/01/1997')) = 1 fetch first 3 rows only	UPDATE ----- 01-NOV-1997 11- 01-NOV-1997 12- 01-NOV-1997 06- 3 record(s) selected.

表 19 の照会と結果は、BIORS.CONTAINS_LE および BIORS.CONTAINS_GE 関数の使用方法を示します。

表 19. BIORS.CONTAINS_LE および BIORS.CONTAINS_GE 関数を使用した照会の例

照会	結果
select s.molweight from Swiss s where biors.CONTAINS_LE(s.molweight, 100368) = 1 fetch first 3 rows only	MOLWEIGHT ----- 100368 10576 8523 3 record(s) selected.
select s.molweight from Swiss s where biors.CONTAINS_GE(s.molweight, 100368) = 1 fetch first 3 rows only	MOLWEIGHT ----- 100368 103625 132801 3 record(s) selected.
select s.journalissue from Swiss s where biors.CONTAINS_GE(s.journalissue, 172) = 1 fetch first 3 rows only	JOURNALISSUE ----- 172 21 242 196 3 record(s) selected.

表 20 の照会と結果は、BioRS 形式を使用した検索語の指定に BIORS.SEARCH_TERM を使用する方法を示します。

表 20. BIORS.SEARCH_TERM 関数を使用した照会の例

照会	結果
<pre>select s.publicationyear from Swiss s where biors.SEARCH_TERM (s.id, '[PublicationYear EQ number:1997;]')=1 fetch first 10 rows only</pre>	<pre>PUBLICATIONYEAR ----- 1997 1997 2000 1988 1991 1997 1994 1997 1997 1998 1994 1995 1997 1997 1999 1997 1994 1994 1995 1993 1992 1997 10 record(s) selected.</pre>
<pre>select s.molweight from Swiss s where biors.SEARCH_TERM (s.id, '[MolWeight EQ number:100368;]') = 1 fetch first 10 rows only</pre>	<pre>MOLWEIGHT ----- 100368 100368 2 record(s) selected.</pre>
<pre>select s.molweight from Swiss s where biors.SEARCH_TERM (s.id, '[MolWeight GREATER number:100368;]') = 1 fetch first 10 rows only</pre>	<pre>MOLWEIGHT ----- 100368 103625 132801 194328 130277 287022 289130 135502 112715 112599 10 record(s) selected.</pre>

以下の照会は、関係述部を使用して、親子関係のある 2 つのデータ・バンク間で等価結合を構成する方法を示します。

```
select s.id, f.id, f.parent from Swiss s, Swissft f
where (f.parent = s.id) fetch first 10 rows only
```

照会結果は以下のようになります。

ID	ID	PARENT
100K_RAT	100K_RAT.1	swissprot:100K_RAT
100K_RAT	100K_RAT.2	swissprot:100K_RAT
100K_RAT	100K_RAT.3	swissprot:100K_RAT
100K_RAT	100K_RAT.4	swissprot:100K_RAT
100K_RAT	100K_RAT.5	swissprot:100K_RAT
100K_RAT	100K_RAT.6	swissprot:100K_RAT
100K_RAT	100K_RAT.7	swissprot:100K_RAT
100K_RAT	100K_RAT.8	swissprot:100K_RAT

```
100K_RAT          100K_RAT.9          swissprot:100K_RAT
104K_THEPA        104K_THEPA.1          swissprot:104K_THEPA
```

10 record(s) selected.

上記の照会結果では、100K_RAT レコードが 9 つの子レコードの親になります (100K_RAT.1 から 100K_RAT.9)。

関連概念:

- 97 ページの『BioRS ラッパー・パフォーマンスの最適化方法』
- 90 ページの『BioRS ラッパーの等価結合述部』

関連資料:

- 87 ページの『カスタム関数と BioRS 照会』
- 86 ページの『CREATE NICKNAME ステートメント - BioRS ラッパーの例』
- 589 ページの『CREATE NICKNAME ステートメント構文 - BioRS ラッパー』

BioRS ラッパー・パフォーマンスの最適化

BioRS ラッパー・パフォーマンスの最適化方法

このトピックでは、BioRS ラッパーを使用した場合の照会のパフォーマンスを最適化する方法を示します。

検索エンジン間の転送データ量を最小化する。

フェデレーテッド環境は 2 つの照会エンジンを使用します。BioRS ラッパーの場合、照会エンジンは DB2[®] Universal Database と BioRS です。DB2 エンジンはニックネーム列に指定された述部 (=、BETWEEN、LIKE、<> などの関係演算子) を処理します。BioRS エンジンは、BioRS ラッパーの 4 つのカスタム関数を使用して指定された述部を処理します。

2 つの検索エンジン間で転送するデータ量を最小化するには、可能な場合は常にデータ処理を BioRS システムにプッシュダウンするよう照会を構成します。

照会で結合操作を行う必要がある場合、可能な場合は BioRS データ・バンクの既存の親子関係を活用して等価結合操作を行います。等価結合操作は BioRS で処理され、ここでも DB2 と BioRS 照会エンジン間のデータ転送量を最小化できます。

重要: DB2 Information Integrator から BioRS への照会を中断しないでください (コマンド行プロセッサで **Ctrl-D** または **Ctrl-Z** を使用する、あるいはアプリケーション・プログラムを停止する、など)。照会を中断すると、BioRS サーバーで実行中の処理がデッド状態になります。「デッド」プロセスは、BioRS と DB2 Information Integrator 両方のシステム・パフォーマンスを急速に低下させます。また、「デッド」プロセスの数が増えると、DB2 Information Integrator 照会処理中に予期しないエラーが発生する恐れがあります。例えば、行があると予想される場合に、有効な照会が 0 行を戻す、などです。極端な状態では、BioRS、DB2 Information Integrator のいずれかまたは両方が停止または異常終了する恐れがあります。

フェデレーテッド環境で BioRS 統計情報を保守する。

フェデレーテッド・システムでは、フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカatalog統計に基づいて、照会処理を最適化します。BioRS データ・ソースに関する現行統計の保守は、BioRS ラッパーのパフォーマンス最適化に不可欠です。ニックネームが定義されたりリモート・オブジェクトの統計データまたは構造の特性を変更した場合、フェデレーテッド・システムで対応するニックネーム列カーディナリティー統計を更新する必要があります。

BioRS ラッパーのパフォーマンスを最適化するには、これらの更新を DB2 Information Integrator で定期的に行う必要があります。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『照会処理のチューニング』
- 90 ページの『BioRS ラッパーの等価結合述部』
- 98 ページの『BioRS 統計情報』

関連資料:

- 87 ページの『カスタム関数と BioRS 照会』
- 92 ページの『BioRS ラッパー - 照会例』

BioRS 統計情報

フェデレーテッド・システムでは、フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカatalog統計に基づいて、照会処理を最適化します。これらの統計は、CREATE NICKNAME ステートメントを使用したニックネーム作成時に BioRS データ・ソースから抽出されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。情報はデータ・ソース・カatalogから読み取られ、フェデレーテッド・サーバー上の DB2[®] フェデレーテッド・データベース・システム・カatalogに置かれます。

BioRS データ・ソースでは、主要な統計情報は以下のものを含まれます。

- ニックネームのカーディナリティー。BioRS データ・ソースでは、ニックネーム・カーディナリティーは対応する BioRS データ・バンクの項目数と同じです。
- BioRS _ID_ エレメントに対応する列のカーディナリティー。この列のカーディナリティーは、列を参照するニックネームのカーディナリティーと一致する必要があります。
- BioRS ラッパーが使用する必要があるすべての列のカーディナリティー。

BioRS データ・ソースに関する現行統計を保守し、BioRS ラッパーのパフォーマンスを最適化してください。ニックネームが定義されたりリモート・オブジェクトの統計データまたは構造の特性を変更した場合、フェデレーテッド・システムで対応するカーディナリティー統計を更新する必要があります。カーディナリティー統計は SYSSTAT.TABLES カatalog・ビューおよび SYSSTAT.COLUMNS カatalog・ビューに保管されます。

フェデレーテッド・システムの BioRS カーディナリティー統計を保守するには、以下のタスクを行ってください。

1. 必要に応じて、必要なニックネームのカーディナリティー統計を確認する。
2. 必要な 1 つまたは複数のカタログ・ビューの適切なカーディナリティー統計を更新する。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『照会処理のチューニング』

関連タスク:

- 99 ページの『BioRS データ・バンク・カーディナリティー統計の確認』
- 99 ページの『BioRS ニックネーム・カーディナリティー統計の更新』
- 100 ページの『BioRS 列カーディナリティー統計』
- 101 ページの『BioRS _ID_ 列カーディナリティーの更新』

BioRS データ・バンク・カーディナリティー統計の確認

ニックネーム統計を更新、または、BioRS _ID_ エレメントに対応する列のカーディナリティーを更新する前に BioRS データ・バンク・カーディナリティー統計を確定する必要があります。

手順:

BioRS 内の特定のデータ・バンクのカーディナリティー統計を確認するには、BioRS ユーティリティー・プログラム `admin_find` または `www_find.cgi` を使用します。-c (カーディナリティー) オプションを指定してください。これら 2 つの BioRS ユーティリティー・プログラムの詳細については、ご使用の BioRS の資料を参照してください。

関連概念:

- 98 ページの『BioRS 統計情報』

関連タスク:

- 99 ページの『BioRS ニックネーム・カーディナリティー統計の更新』
- 100 ページの『BioRS 列カーディナリティー統計』
- 101 ページの『BioRS _ID_ 列カーディナリティーの更新』

BioRS ニックネーム・カーディナリティー統計の更新

ニックネームを作成する BioRS データ・バンクのコンテンツを大幅に変更する場合、BioRS ニックネーム・カーディナリティー統計を更新する必要があります。ニックネームの正しいカーディナリティー統計を維持することで、オプティマイザーと BioRS ラッパーが最適なデータ・アクセス・プランを選択できます。

BioRS ニックネーム・カーディナリティー統計を更新するには、SYSSTAT.TABLES カタログ・ビューを正しいカーディナリティー番号で変更してください。

前提条件:

更新する統計のニックネームに対応する BioRS データ・バンクのカーディナリティー番号を判別してください。

手順:

以下の構文を使用して UPDATE ステートメントを発行してください。

```
UPDATE sysstat.tables SET card=cardinality
WHERE tabschema=nickname-schema
AND tabname=nickname-name;
```

- *cardinality* は、更新対象の統計を持つニックネームに対応する BioRS データ・バンク・カーディナリティー番号です。
- *nickname-schema* は、更新対象の統計を持つニックネームに関連するスキーマの名前です。
- *nickname-name* は、更新対象の統計を持つニックネームの名前です。

関連概念:

- 98 ページの『BioRS 統計情報』

関連タスク:

- 99 ページの『BioRS データ・バンク・カーディナリティー統計の確認』
- 100 ページの『BioRS 列カーディナリティー統計』
- 101 ページの『BioRS _ID_ 列カーディナリティーの更新』

BioRS 列カーディナリティー統計

フェデレーテッド・システムの BioRS 列カーディナリティー統計を更新するには、SYSSTAT.COLUMNS カタログ・ビューを変更する必要があります。

BioRS 列の正しいカーディナリティー統計を維持することで、オプティマイザーと BioRS ラッパーが照会処理中に最適なデータ・アクセス・プランを選択できます。

BioRS をフェデレーテッド・システムに追加する作業の一部として BioRS 列カーディナリティーをオプションで更新できます。BioRS データ・ソースへの照会のパフォーマンスを向上させたい場合にも BioRS 列カーディナリティーを更新します。

制約事項:

この手順は、BioRS _ID_ エレメントに対応する列のカーディナリティー統計の更新には使用しないでください。BioRS _ID_ エレメントに対応する列のカーディナリティー統計を更新する場合は、別の手順で行う必要があります。

手順:

BioRS 列カーディナリティー統計を更新するには、以下の構文を使用して UPDATE ステートメントを発行してください。

```
UPDATE sysstat.columns SET colcard=(SELECT COUNT(DISTINCT column-name)
FROM nickname-schema.nickname-name)
WHERE
  tabschema=nickname-schema
  AND tabname=nickname-name
  AND colname=column-name;
```

- *column-name* は、更新対象のカーディナリティー統計を持つ列の名前です。
- *nickname-schema* は、指定列を使用するニックネームに関連するスキーマの名前です。

- *nickname-name* は、指定列を使用するニックネームの名前です。

ニックネームに指定されたデータ・バンクの項目をすべて検索するため、照会の実行には数分かかる場合があります。

列に複数の値を含められる場合 (SwissProt データベース形式の `PublicationYear` エレメントなど)、SQL 参照を使用するには計算が複雑になり過ぎます。このような列には、カーディナリティー値を手動で計算後、`SYSSTAT.COLUMNS` カタログ・ビューを更新する必要があります。カーディナリティー値を計算するには、列の別個の値を、行ごとの値の平均値で割ります。計算したカーディナリティー値は、表のカーディナリティーより小さくなります。

例:

3 つの行にニックネームがあるとします。それぞれの行に対する `PublicationYear` 列の値は、以下のとおりです。

- 1997 1992 1985
- 1997 1992 1982
- 1992 1991 1990 1976 1974 1971

9 つの異なる値があり、行内の値の平均数は 4 です。`PublicationYear` 列のカーディナリティーは 9/4 または 3 (2.25 は次に大きい整数に切り上げ) です。カーディナリティーの計算後、`SYSSTAT.COLUMNS` カタログ・ビューを次の `UPDATE` ステートメントを使用して更新します。

```
UPDATE sysstat.columns SET colcard=3
WHERE
    tabschema=nickname-schema
    AND tabname=nickname-name
    AND colname=column-name
```

- 3 は列のカーディナリティー値です。
- *nickname-schema* は、指定列を使用する基本ニックネームに関連するスキーマの名前です。
- *nickname-name* は、指定列を使用する基本ニックネームの名前です。
- *column-name* は、更新対象のカーディナリティー統計を持つ列の名前です。

関連概念:

- 98 ページの『BioRS 統計情報』

関連タスク:

- 99 ページの『BioRS ニックネーム・カーディナリティー統計の更新』
- 101 ページの『BioRS `_ID_` 列カーディナリティーの更新』

BioRS `_ID_` 列カーディナリティーの更新

BioRS `_ID_` エレメントにマップする列の正しいカーディナリティー統計を維持することで、オプティマイザーと BioRS ラッパーが最適なデータ・アクセス・プランを選択できます。

BioRS `_ID_` エレメントにマップする列のカーディナリティー番号を更新するには、`SYSSTAT.COLUMNS` カタログ・ビューを変更する必要があります。

前提条件:

列を参照するニックネームに対応する BioRS データ・バンクのカーディナリティー番号を判別してください。BioRS _ID_ エlementにマップする列のカーディナリティー番号は、列を参照するニックネームのカーディナリティーと一致する必要があります。

手順:

BioRS _ID_ 列カーディナリティー統計を更新するには、以下の構文を使用して UPDATE ステートメントを発行してください。

```
UPDATE sysstat.columns SET colcard=<cardinality>
WHERE
  tabschema=nickname-schema
  AND tablename=nickname-name
  AND colname IN (SELECT colname FROM syscat.coloptions
                  WHERE
                    tabschema=nickname-name
                    AND tablename=nickname-name
                    AND option='ELEMENT_NAME';
                  AND setting='_ID_')
```

- *cardinality* は、列のニックネームに対応する BioRS データ・バンク・カーディナリティー番号です。
- *nickname-schema* は、列のニックネームに関連するスキーマの名前です。
- *nickname-name* は、列を使用するニックネームの名前です。

関連概念:

- 98 ページの『BioRS 統計情報』

関連タスク:

- 99 ページの『BioRS データ・バンク・カーディナリティー統計の確認』
- 99 ページの『BioRS ニックネーム・カーディナリティー統計の更新』
- 100 ページの『BioRS 列カーディナリティー統計』

BioRS ラッパーのメッセージ

このトピックでは、BioRS ラッパーの使用時に表示される可能性のあるメッセージについて説明します。

表 21. BioRS のラッパーが発行するメッセージ

エラー・コード	メッセージ	説明
SQL0604N	列の長さ、精度、または位取り属性、特殊タイプ、構造化タイプ、構造化タイプの属性、関数、あるいはタイプ・マッピング <data-item> が無効です。	ニックネーム列のデータ・タイプが、基本データ・バンク・Elementの BioRS タイプと互換性がありません。CREATE NICKNAME ステートメント内の列のデータ・タイプをチェックしてください。

表 21. BioRS のラッパーが発行するメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由「ラッパー・オブジェクトの作成エラー」。)	新しいラッパー・オブジェクトの作成時にエラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "BioRS <trace-point></code>.")	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "Memory allocation failed: <trace-point>.")	メモリー割り振りの際に、エラーが発生しました。フェデレーテッド・サーバーのホストに十分なメモリーがあるか確認し、照会を再度サブミットしてください。問題が解決しない場合は、IBM ソフトウェア・サポートに連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。(理由 "sqlno_crule_save_plans[100]:rc(-214272209) Empty plan list.")	オブティマイザー・プログラムと BioRS ラッパーが照会実行プランで一致しません。照会を単純化し、再実行してください。
SQL0401N	演算 "=" のオペランドのデータ・タイプが一致していません。	照会が無効です。カスタム関数述部の右側の式は整数値でなければなりません。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "" を受け取りました。関連したテキストとトークンは "Databank not found" です。	CREATE NICKNAME ステートメントが参照する BioRS データ・バンクが BioRS サーバーで見つかりません。CREATE NICKNAME ステートメントをチェックし、参照先のデータ・バンク名が正しいか確認してください。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "" を受け取りました。関連したテキストとトークンは "Connection timed out" です。	BioRS サーバーが TIMEOUT オプションが指定した時間内に通信要求に応答できませんでした。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連したテキストとトークンは "Error reading from server" です。	BioRS サーバーからのデータ読み取り中に通信エラーが発生しました。エラーの詳細については、<trace_point> エラー・コードの値を確認してください。

表 21. BioRS のラッパーが発行するメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連したテキストとトークンは "Host not found" です。	HOST サーバー・オプションで指定した BioRS サーバー・ホストが見つかりません。CREATE SERVER ステートメントをチェックし、HOST サーバー・オプションの値が正しいか確認してください。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連したテキストとトークンは "Unable to connect to server" です。	ラッパーは HOST サーバー・オプションが示すサーバーに接続できませんでした。エラーの詳細については、<trace_point> エラー・コードの値を確認してください。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連したテキストとトークンは "Unable to create TCPIP socket" です。	ラッパーは TCPIP ソケットを作成できませんでした。エラー・コードの詳細については、<trace_point> エラー・コードの値を確認してください。
SQL1822N	データ・ソース "BioRS wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連したテキストとトークンは "Error sending to server" です。	ラッパーは要求を BioRS サーバーに送信できませんでした。エラーの詳細については、<trace_point> エラー・コードの値を確認してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Cannot change case-sensitivity of server"。	SQL ステートメントで CASE_SENSITIVE サーバー・オプションの値は変更できません。このオプションの値を変更するには、サーバーをドロップする必要があります。その後、CREATE SERVER ステートメントを使用してサーバーを再作成し、CASE_SENSITIVE オプションに正しい値を指定する必要があります。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Multiple joins between two nicknames"。	2 つのニックネーム間では 1 つの結合述部のみ可能なため、照会が無効です。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Right side of function predicate must be constant"。	照会が無効です。カスタム関数述部の右側の式は定数でなければなりません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Arg 1 of custom function not a column"。	照会が無効です。カスタム関数の最初の引き数は BioRS ニックネームの列を参照する必要があります。

表 21. BioRS のラッパーが発行するメッセージ (続き)

エラー・コード	メッセージ	説明
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Arg 1 of CONTAINS function not indexed".	照会が無効です。 BIORS.CONTAINS、 BIORS.CONTAINS_LE、または BIORS.CONTAINS_GE 関数の最初の引き数が参照する列は索引付き列である必要があります。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Bad type for arg1 of <function-name> function."	照会が無効です。 BIORS.CONTAINS、 BIORS.CONTAINS_LE、または BIORS.CONTAINS_GE 関数の最初の引き数が参照する列が、不正なデータ・タイプです。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Arg 1 of SEARCH_TERM not _ID_ column".	照会が無効です。 SEARCH_TERM 関数の最初の引き数が参照する列が BioRS _ID_ をマップしません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Bind parameter cannot be NULL".	BIORS.CONTAINS 関数の 2 番目の引き数が参照する列またはホスト変数が NULL です。BioRS ラッパーは NULL 値を処理できません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Cannot convert value to BioRS literal".	値はリテラル、列、またはホスト変数でラッパーにサブミットされましたが、有効な BioRS リテラルに変換できませんでした。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Cannot change server version".	ALTER SERVER ステートメントでサーバー・バージョンは変更できません。サーバー・バージョンを変更するには、サーバーをドロップする必要があります。その後、CREATE SERVER ステートメントを使用してサーバーを現行バージョンに再作成してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Bad type for arg2 of <function-name> function".	照会が無効です。 BIORS.CONTAINS、 BIORS.CONTAINS_LE、または BIORS.CONTAINS_GE 関数の 2 番目の引き数が参照する列が、不正なデータ・タイプです。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = "Nickname has no columns".	CREATE NICKNAME ステートメントに列宣言が指定されていません。ニックネームの作成時には、列宣言が必要です。

関連概念:

- | • 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』
- | **関連資料:**
- | • 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 7 章 BLAST データ・ソースへのアクセスの構成

この章では、BLAST データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。BLAST データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- BLAST の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- BLAST ラッパーに関連するエラー・メッセージをリストします。

BLAST とは?

BLAST (Basic Local Alignment Search Tool) は、National Center for Biotechnology Information (NCBI) によってメンテナンスされているユーティリティです。BLAST は、ヌクレオチドまたはアミノ酸のシーケンス・データベースをスキャンして「ヒット」を探すために使用されます。1 つの BLAST ヒットには 1 つまたは複数の高スコア・セグメント・ペア (HSP) が含まれます。HSP は、配列が局所的に最大で、類似性スコアが一定のしきい値を超える、シーケンス・フラグメントのペアです。NCBI は、実行可能ファイル blastall を提供しています。これは GenBank や SWISS-PROT などの BLAST 可能なデータ・ソースで BLAST 検索を実行するために使用されます。

BLAST ラッパーは、BLAST 検索の 5 つのタイプすべて、すなわち BLASTn、BLASTp、BLASTx、tBLASTn、および tBLASTx をサポートします。これらの内容は、表 22 に示されています。

表 22. BLAST ラッパーによってサポートされている BLAST 検索タイプ

BLAST 検索タイプ	説明
BLASTn	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同種の領域をもつシーケンスを検出します。
BLASTp	BLAST 検索の 1 つのタイプ。アミノ酸シーケンスをアミノ酸シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同種の領域をもつシーケンスを検出します。
BLASTx	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをアミノ酸シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同種の領域をもつシーケンスを検出します。照会シーケンスは 6 つの読み取りフレームすべてで変換されます。結果の各シーケンスは、シーケンス・データベースを検索するために使用されます。

表 22. BLAST ラッパーによってサポートされている BLAST 検索タイプ (続き)

BLAST 検索タイプ	説明
tBLASTn	BLAST 検索の 1 つのタイプ。アミノ酸シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同種の領域をもつシーケンスを検出します。シーケンス・データベースの中のシーケンスは 6 つの読み取りフレームすべてで変換されます。結果のシーケンスは、照会シーケンスの領域と同種の領域を検出するために検索されます。
tBLASTx	BLAST 検索の 1 つのタイプ。ヌクレオチド・シーケンスをヌクレオチド・シーケンス・データベースの内容と比較して、オリジナルのシーケンスの領域と同種の領域をもつシーケンスを検出します。 tBLASTx 検索では照会シーケンスとシーケンス・データベースの両方とも 6 つの読み取りフレームすべてで変換されます。結果のシーケンスは、同種の領域を検出するために比較されます。

図 7 は BLAST が実際のフェデレーテッド・システムでどう機能するかを示しています。

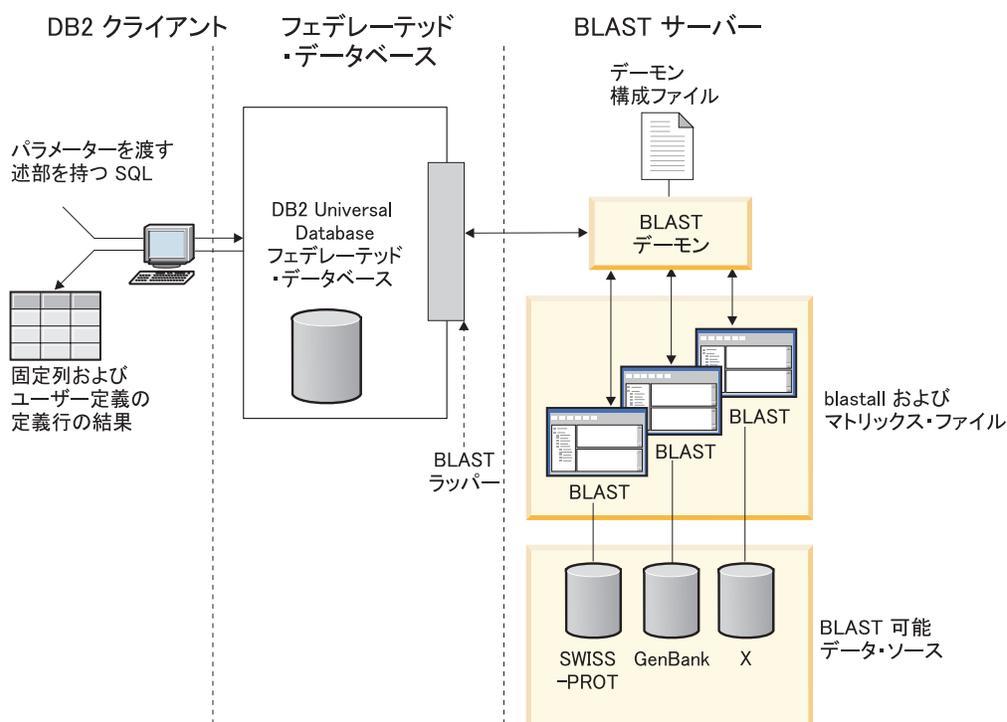


図 7. BLAST ラッパーがどのように機能するか

クライアント・サイドで、ユーザーまたはアプリケーションは、標準の BLAST オプションにマップする BLAST 特有のパラメータを渡す述部を持つ SQL ステートメントをサブミットします。入力述部を持つ SQL ステートメントは、BLAST ラッパーがインストールされた DB2® Universal Database フェデレーテッド・データベース・システムへ送信されます。

BLAST ラッパーは、照会を BLAST アプリケーションが認識できるフォーマットに変換し、その変換された照会を BLAST サーバーへ送信します。このサーバーは

フェデレーテッド・システムのマシンとは別のマシンであってもかまいません。特別なデーモン・プログラムが BLAST サーバーで実行されます。このデーモンは、デーモン構成ファイルからの情報を使用して、フェデレーテッド・システムからの照会要求を受信し、それを BLAST アプリケーションへ送信します。次に BLAST アプリケーションは、BLAST 可能なデータ・ソースに対して通常の方法で実行されます。

結果は BLAST へ戻され、次にデーモンへ戻されます。デーモンは検索されたデータを BLAST ラッパーへ戻します。ラッパーはデータをリレーショナル表フォーマットへ変換し、この表をユーザーまたはアプリケーションへ戻します。戻されたデータは以下の 2 つの部分で構成されています。

- BLAST ユーザーが見なれている一連の標準の固定列
- ユーザー構成の定義行情報

以下の例は、BLAST 可能なデータ・ソースからリレーショナル情報を抽出する方法を示しています。データは、未加工の fasta ファイル・フォーマットから BLAST 可能なデータ・セットへ、そして実際のフェデレーテッド・システムで他のデータ・ソースと結合可能なリレーショナル表へと移動します。

110 ページの図 8 は fasta ファイルのサンプルで、4 つの定義行およびヌクレオチド・シーケンス・レコードが含まれています。

```

>7:4986 PMON5744
GTTCTTCCCAGTGCCCAAGTCCATTCTGACATCAATGAAGAAGGTAATAATCCCTGCGTGATCCCTCTGCC
AAGATGTCGAAATCAGACCCGGATAAACTAGCTGCTGTCAGAATAACAGACAGCCCGGAGGAGATCGTGC
AGAAGTTCGCAAGGCTGTGACGGACTTACCTCGGAGGTACCTACGACCCGGCCAGGCGAGGAGCGGT
GTCCAACCTGGTGGCCATCCACGCGGCAGTGACCGGACTCCCGGTGGAGGAGGTGGTCCGCCAAGTGCT
GGCATCAACACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTGCACCAATTAAGA
GTGAAATTTGAAAACTGAAGAGGAACAAGGACCACCTAGAGAAGGTTTTACAAGTTGGGTCGGCAAAAGC
CAAAGAATTAGCATATCCCGTGTGCCAGGAGGTGAAGAAATGGTGGGGTTTCTATAGGCAGTCTCACCT
AGTCCCAGAAAATGTTTTTATCTTGTGGTCTGCTTGCACACTCAGTCTAATAAAGGCAGCTTTCCTAAG
ACGCCAACAATTCAGTTTGGGGATGCTTAGTTACT

>8:9747 PMON5699
AAGAAGTTCTTGTAGAACTTCCACCTCCGGCTTCCCTCCACCTCTCTACTGTCCCAACCTTCTGAG
ACGCTTTTTCTCTCCCGAGGATTTATCTTTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCTCT
TTTTCCCCCGAGGCTGGTTTTGCTTTGGGAGGGGGGTTTTTTAAAGGGCCGGGGGGCCCCCTTT
CTCCCCCTAATGGGGTTAATTAATAATGGGGGGGGGGTTTTTTTTTTTTTAAACCCCTATTTGGTCCGG
CCCGGGGATTTCCCCCCCCCCCCCTTCCCGGTTCCGGGGCCGGAGGAGGGGGGAAAAGGGCGGGAA
CCTTTGGTAGTTTCCCTCGGAAAAAATTTTTCGGGGGGAAAACCTCCCT

>13:6512 PMON5498
GATAAGAGGCAGAATAGAAGACTGGACTACTTCTCTCCTAAAAACACATTTAAAACCTAAGCCTGAGCAAT
CTCCACCAAATGGACCGAAACCTTAAAAAGAATCCTACTCCTGAAGAAAAAGAGGAGGACACATCAA
GAGGTAGAAGGGCGATTTATGATATAAACACCCCATACCTCCAGAGTGGGAAGCTCCACAGACTGAA
AACTAACTGGTTCACAGAAACTCACCTACAGGAGTGAGCCCCACATCAAACCTCGAATGTGGGGATCTG
GCACTGGTAGAAAAGCCCTGGAGCATCTGGCATTGAAGCCAGTGGGGCTTGTGTGCAGGAGATCCAC
AGGACTAGGGGAAACGGAGACCCCTTCTTAAAAGGTGCACACAGACTTTTACGTGCACTGGGTCCAG
TGCAAAGCAAAGTCTCCATAGGAATCTGGGTCAAACCTGACTGCAGTTCTTGGAGGACCTCTGGGAAAG
CAAGGGTGAATGTGGCTTCTTGTGGGAAAGGACATTGGAAGCAAAGCTTGGGAATATTCATCAGTGT
GC

>15:8924 PMON5426
GGAGAACTGACTCCTGAGCAGCTGCAATTCATGCGGCAGGTGCAGCTCGCCCAGTGGCAGAAGACGCTG
CCACAGCGCGGACCCGGAACATCGTGACCGGCTGGGCATCGGGGCGCTGGTGTGGCAATTTGTATCC
GTTTGGAGCTGTAGACTCAGGGAGACCGCATTTAGGGGAACAGGAAGGGCAGCAGGGGCGTGTAGGAGGGC
AGTGTGGGGTGGTAGAAGGAGCCCGAGATATGAAAACCTTGGCTCCTTTAACTTGAATCAAGCGTTT
GGTGTACCTTACGTTGTCAATTTAAAGGTATTTTGTATATAATTGATTAATGATTACGGAGTCGGGTGA
GGGCTCCAGGAGCAGACGGCAGAAGATCGAATTTGGGAGGATGATCAGCAGCGGTGGTTGAGCAAGTGT
GGGAAAAGGGAATGCGCACATTCACGTTGTTTCTGAACCCACCTCCCAGATGGTTACACCTTCTACT
CGGTGTCCAGGAGCGTTTCTTGGATGAGCTGGAGGATGAGGCCAAAGCTGCTC

```

図8. サンプルの *fasta* ファイル *nucleo1*

標準の `formatdb` アプリケーションが *fasta* ファイルを BLAST 可能なデータ・セットへ変換します。データは、BLAST ラッパーがインストールされ、登録されているフェデレーテッド・システムを通じて、SQL によって照会できる状態になっています。

クライアント・エンドにおいてユーザーまたはアプリケーションによって送信された以下の照会は、BLAST ラッパーによって変換されます。そして、BLAST 可能なデータ・セットに対して実行されます。

```

SELECT Unique_ID, Experiment_Number, Organism_Number, HSP_Info, Score
FROM nucleo1
WHERE BlastSeq = 'ACATTCCTATAGAGTATTGCTACTCCTCCAGGATAGAGTCATCTCT
GGTCTCCAGAGCCACCGCTGGCTACAAGTTGGTGGTGGCGGAGGCTGTGATTGAGAGATTTG
CACCAATACAGAACTCACCTACAGGAGTGAGCGGGTGGTGAAGGAGCCCGAGATATGAAA
ACCTTGTTCAGACCCATTGTCACCGGGG';

```

照会の結果は、BLAST ラッパーによって 111 ページの表 23 に示されているリレーショナル表フォーマットに変換されます。

表 23. フェデレーテッド・システムに組み込まれると、BLAST は、結果をリレーショナル表フォームで戻します。

固有な ID	実験番号	生体番号	HSP_INFO	スコア
PMON5744	4986	7	一致 = 57/201 (28%)、ポジティブ = 57/201 (28%)、ギャッ プ = 0/201 (0%)	+1.13487000000000E+002
PMON5426	8924	15	一致 = 35/201 (17%)、ポジティブ = 35/201 (17%)、ギャッ プ = 0/201 (0%)	+6.98754000000000E+001
PMON5498	6512	13	一致 = 26/201 (13%)、ポジティブ = 26/201 (13%)、ギャッ プ = 0/201 (0%)	+5.20342000000000E+001

データは完全にリレーショナル・フォームになっており、研究所で使用される他のデータ・ソースのデータと結合できます。いくつかのデータ・ソースの結果を結合することによって、フェデレーテッド・システムのインプリメンテーション前には容易にまたは効率的に発見できなかつたような点を洞察することができます。

関連タスク:

- 111 ページの『フェデレーテッド・サーバーへの BLAST データ・ソースの追加』

フェデレーテッド・サーバーへの BLAST の追加

フェデレーテッド・サーバーへの BLAST データ・ソースの追加

フェデレーテッド・サーバーが BLAST データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、フェデレーテッド・サーバーが BLAST データ・ソースにアクセスするように構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

BLAST データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. blastall 実行可能ファイルおよびマトリックス・ファイルの正しいバージョンがインストールされていることを検証します。
2. BLAST デーモンを構成します。
3. BLAST デーモンを開始します。
4. ラッパーを登録します。
5. サーバー定義を登録します。
6. BLAST 検索のニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 112 ページの『blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証』
- 113 ページの『BLAST デーモンの構成』
- 117 ページの『BLAST デーモンの開始』
- 119 ページの『BLAST ラッパーの登録』
- 121 ページの『BLAST データ・ソースのサーバーの登録』
- 121 ページの『BLAST データ・ソースのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証

blastall 実行可能ファイルとマトリックス・ファイルの正しいバージョンがインストールされていることの検証は、BLAST をフェデレーテッド・システムに追加するという、より大きな作業の一部を成します。

前提条件:

blastall 実行可能ファイル、および BLOSUM62、BLOSUM80、PAM30、PAM70 マトリックス・ファイルの最新バージョンが、使用する BLAST サーバー・マシンにインストール済みであることを確認します。blastall 実行可能ファイルの最新バージョンがない場合は、バイナリー・ファイルとマトリックス・ファイルをインストールする必要があります。マトリックス・ファイルは、blastall 実行可能ファイルと同じディレクトリーになければなりません。

手順:

blastall 実行可能ファイルとマトリックス・ファイルのバージョン・レベルのチェックは、次のように行います。

1. コマンド行から BLAST 探索を実行し、出力ファイルに現れるバージョン番号に注目します。
2. この製品の Web サイトで、このラッパーを使用してテストした BLAST のバージョンをチェックし、サポートされているバージョンであることを確認してください。

この一連の作業における次のタスクは、『BLAST デーモンの構成』です。

関連タスク:

- 113 ページの『BLAST デーモンの構成』

BLAST デーモンの構成

BLAST デーモンの構成は、BLAST をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

BLAST ラッパーには BLAST デーモンが必要です。BLAST デーモンは、フェデレーテッド・システムから TCP/IP を介してアクセス可能なサーバー上で実行していなければなりません。このサーバーは、フェデレーテッド・サーバーとして機能するものと同じサーバーにすることも、別の BLAST サーバーにすることもできます。

デーモンは、ラッパーおよびフェデレーテッド・データベースとは別個に実行します。デーモンは、ラッパーからの BLAST ジョブ要求を listen します。

前提条件:

BLAST デーモンは以下の権限を持っている必要があります。

- BLAST 検索の実行のため、blastall バイナリー・ファイルに対する実行アクセス権限。
- 一時ファイルに書き込むことができるディレクトリーに対する書き込みアクセス権限。
- BLAST 検索を実行できる対象となる、少なくとも 1 つの BLAST 可能データ・ソースに対する読み取りアクセス権限。blastall 実行可能ファイルは、formatdb プログラムによって生成されたデータ・ファイルと BLAST 索引ファイルの両方に対する読み取りアクセス権限をもっている必要があります。

制約事項:

実行可能ファイルまたはデータベース・パスにスペースが含まれる場合、BLAST デーモンは正常に実行しないことがあります。例えば、BLAST 実行可能ファイルを Windows サーバー上の C:%Program Files にはインストールしないでください。

手順:

BLAST デーモンを構成するには、以下の手順に従います。

1. BLAST デーモンの実行可能ファイルが適切なサーバー上に存在することを確認します。DB2 Information Integrator のインストール時に、デーモン実行可能ファイルはフェデレーテッド・サーバー上の以下のディレクトリーにインストールされます。

UNIX の場合

デーモン実行可能ファイルは `db2blast_daemon` です。このファイルは、`$DB2PATH/bin` ディレクトリーにインストールされます。

Windows の場合

デーモン実行可能ファイルは、`db2blast_daemon.exe` および `db2blast_daemon_svc.exe` です。これらのファイルは、`%DB2PATH%\bin` ディレクトリーにインストールされています。

別個の BLAST サーバー・コンピューターを使用する場合、デーモン実行可能ファイルをフェデレーテッド・サーバー上のディレクトリーから BLAST サーバー・コンピューター上のファイルにコピーする必要があります。デーモン実行可能ファイルは、ディレクトリー・パス内の名前にスペースを含まない、BLAST サーバー・コンピューター上の任意のディレクトリーで実行できます。

2. BLAST デーモンの構成ファイルが適切なサーバー上に存在することを確認します。DB2 Information Integrator のインストール中に、サンプルのデーモン構成ファイル `BLAST_DAEMON.config` がフェデレーテッド・サーバー上のディレクトリーにインストールされます。

UNIX の場合

デーモン構成ファイルは、`$DB2PATH/bin` ディレクトリーにインストールされます。`$DB2PATH` は、DB2 Information Integrator がインストールされているディレクトリーです。

Windows の場合

デーモン構成ファイルは、`%DB2PATH%\bin` ディレクトリーにインストールされます。`%DB2PATH%` は、DB2 Information Integrator がインストールされているディレクトリー (通常は `C:\SQLLIB\bin`) です。

デフォルトでは、デーモンは、デーモンが開始される作業ディレクトリー内で構成ファイルを検索します。構成ファイルは、別のロケーションにコピーすることができます。BLAST サーバー・コンピューターを使用する場合、デーモン構成ファイルをフェデレーテッド・サーバー上のディレクトリーから BLAST サーバー・コンピューター上のファイルにコピーする必要があります。デーモン構成ファイルは、デーモンがアクセス可能な BLAST サーバー・コンピューター上の任意のディレクトリーにコピーできます。

3. デーモン構成ファイルを編集して、データ・ソースを扱うことができるようにします。構成ファイルを名前変更することもできます。
 - 構成ファイルの 1 行目は、等号でなければなりません。等号がないと、デーモンは開始しません。 `DAEMON_PORT` が指定されなかったことを示すエラー・メッセージが出されます。
 - 構成ファイルの最終行は、改行で終了する必要があります。DB2 Information Integrator に付属のサンプル構成ファイルは、改行で終了しています。ファイルを編集するときは、そのファイルの最終行が改行で終了していることを確認

する必要があります。最終行が改行で終わっていない場合、最終行にリストされたデータ・ソースを使用して BLAST 照会を初めて実行しようとする際に、エラー・メッセージが出されます。

- 構成ファイルに以下のオプションを指定します。パスを必要とするオプションについては、相対パスを指定できます。相対パスは、デーモン・プロセスが開始されたディレクトリーに相対しています。

DAEMON_PORT

これは、ラッパーによってサブミットされた BLAST ジョブ要求をデーモンが listen するネットワーク・ポートです。

MAX_PENDING_REQUESTS

これは、任意の一時点でデーモンでブロッキングできる BLAST ジョブ要求の最大数です。この数は、並行して実行されている BLAST ジョブ数を表すものではありません。一時点でブロックできるジョブ要求数のみです。これを 5 より大きい数に設定することをお勧めします。

BLAST デーモンでは、並行して実行できる BLAST ジョブ数の制限はありません。

DAEMON_LOGFILE_DIR

これは、デーモンがそのログ・ファイルを作成するディレクトリーです。このファイルには、BLAST デーモンが生成する有用な状況情報とエラー情報が含まれます。

Q_SEQ_DIR_PATH

これは、デーモンによって一時照会シーケンス・データ・ファイルが作成されるディレクトリーです。この一時ファイルは、BLAST ジョブが完了するとクリーンアップされます。

BLAST_OUT_DIR_PATH

これは、デーモンが BLAST 出力データを保管するための一時ファイルを作成するディレクトリーです。データはこのファイルから読み取られて、ネットワーク接続を介してラッパーに戻されます。データがラッパーに渡されると、デーモンは一時ファイルをクリーンアップします。

BLASTALL_PATH

これは、デーモンを実行しているコンピューター上の BLAST 実行可能ファイルの完全修飾名です。

データベース指定項目

BLAST 可能データ・ソースの場所を指定します。構成ファイルに指定したデータベース *data_source_name* をメモしてください。デーモンが正しく機能するためには、データ・ソースのニックネームを作成するときに、データベース *data_source_name* を指定する必要があります。この名前は、大文字小文字が区別されます。データベース *data_source_name* は、CREATE NICKNAME ステートメントの DATASOURCE オプションに指定されます。

構成ファイルには、少なくとも 1 つのデータベース指定項目が、次のフォームで含まれている必要があります。

```
data_source_name = path to BLAST-able_data_source
```

UNIX の場合

たとえば、GenBank BLAST 可能データ・ソースを指定するには、デーモン構成ファイルに次に示す行を追加します。

```
genbank=/dsk/1/nuc1_data/genbank
```

Windows の場合

たとえば、GenBank BLAST 可能データ・ソースを指定するには、デーモン構成ファイルに次に示す行を追加します。

```
c:%vnr_data%genbank_nonest1.fasta
```

データベース指定項目に示されるパスには、3 つの索引ファイルが含まれていなければなりません。

- ヌクレオチド・データ・ソースの場合、索引ファイルは以下の拡張子をもっています。
 - .nhr
 - .nin
 - .nsq
- アミノ酸データ・ソースの場合、索引ファイルは以下の拡張子をもっています。
 - .phr
 - .pin
 - .psq

データベース指定項目は、オリジナルの Fasta フォーマットされたデータを含んでいるファイルの名前を示していなければなりません。3 つの索引ファイルは、オリジナルの Fasta フォーマットされたデータを含んでいるファイルと同じルート名をもっていなければなりません。

この一連の作業における次のタスクは、『BLAST デーモンの開始』です。

関連タスク:

- 117 ページの『BLAST デーモンの開始』

関連資料:

- 116 ページの『BLAST デーモンの構成ファイル - 例』

BLAST デーモンの構成ファイル - 例

以下の例は、サンプルの構成ファイルの内容を示しています。

例 - BLAST デーモンの構成ファイル (UNIX):

この例は、必須オプションと、GenBank および SWISS-PROT についての BLAST 対応データ・ソースの仕様を示しています。

```
=  
DAEMON_PORT=4007  
MAX_PENDING_REQUESTS=10  
DAEMON_LOGFILE_DIR=./  
Q_SEQ_DIR_PATH=./
```

```
BLAST_OUT_DIR_PATH=./
BLASTALL_PATH=./blastall
genbank=/dsk/1/nuc1_data/genbank
swissprot=/dsk/1/prot_data/swissprot
```

例 - BLAST デーモンの構成ファイル (Windows):

この例は、必須オプションと、GenBank および SWISS-PROT についての BLAST 対応データ・ソースの仕様を示しています。

```
=
|
| DAEMON_PORT=4007
|
| MAX_PENDING_REQUESTS=10
|
| DAEMON_LOGFILE_DIR=%
|
| Q_SEQ_DIR_PATH=%
|
| BLAST_OUT_DIR_PATH=%
|
| BLASTALL_PATH=%blastall.exe
|
| genbank=c:%vnr_data%genbank_nonest1.fasta
|
| swissprot=c:%vnr_data%swissprot
```

関連タスク:

- 111 ページの『フェデレーテッド・サーバーへの BLAST データ・ソースの追加』
- 113 ページの『BLAST デーモンの構成』
- 117 ページの『BLAST デーモンの開始』

BLAST デーモンの開始

BLAST デーモンの開始は、BLAST をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。BLAST データ・ソースへアクセスできるためには、BLAST デーモンを開始しなければなりません。

前提条件:

BLAST デーモンを開始するには、構成ファイルの中の DAEMON_LOGFILE_DIR、BLAST_OUT_DIR_PATH、および Q_SEQ_DIR_PATH 項目の下にリストされているすべてのパスに対する書き込みアクセス権限をもっている必要があります。

手順:

BLAST デーモンを UNIX サーバー・コンピューターで開始するには、以下のようになります。

1. デーモン実行可能ファイルが存在するディレクトリーを開きます。
2. db2blast_daemon コマンドを実行します。
 - デーモン構成ファイルの名前を変更しておらず、構成ファイルがデーモン実行可能ファイルと同じディレクトリーにある場合、コマンド行に以下のコマンドを入力します。

```
db2blast_daemon
```
 - デーモン構成ファイルの名前を変更してしまっているか、構成ファイルがデーモン実行可能ファイルと同じディレクトリーにない場合、ラッパー・デーモン・コマンドで `-c` オプションを使用して、デーモン実行可能ファイルが新しい名前または新しい場所を指すようにする必要があります。

たとえば、以下のコマンドは、ラッパー・デーモンがデーモン構成情報をサブディレクトリー `cfg` の中の `BLAST_D.config` というファイルの中で探すようにさせます。

```
db2blast_daemon -c cfg/BLAST_D.config
```

実行可能ファイルは、BLAST デーモンが実行される新規プロセスを開始します。

BLAST デーモンを Windows サーバー・コンピュータで開始するには、以下のようになります。

1. デーモン実行可能ファイルが存在するディレクトリーを開きます。
2. 必要なパラメーターを指定して、`db2blast_daemon` コマンドを発行します。例えば、デバッグをオンにしてデーモン・サービスをインストールするには、次のようになります。

```
db2blast_daemon -a install -d 2
```

```
db2blast_daemon -a start
```

この一連の作業における次のタスクは、『BLAST ラッパーの登録』です。

関連タスク:

- 119 ページの『BLAST ラッパーの登録』

db2blast_daemon コマンド - 構文および例

| `db2blast_daemon` コマンドは UNIX および Windows サーバーで使用できます。構
| 文内でリストされる引き数の一部は、Windows サーバーでのみ使用できます。

`db2blast_daemon` コマンドの構文は以下のとおりです。

```
db2blast_daemon -a action -c config_file -d debug_level  
-u user_id -p password
```

-a *action*

指定したアクティビティーを実行します。有効なアクションは、*status*、*install*、*start*、*stop*、および *remove* です。

| この引き数は、Windows サーバーでのみ指定できます。

-c *config_file*

指定した構成ファイルを使用するように、デーモン・サービスに指示します。構成ファイルを指定しないと、デーモンはデーモンの実行可能ファイルがインストールされているディレクトリー内で、`BLAST_DAEMON.config` ファイルを検索します。このオプションは、*install* および *start* アクションとともに使用できます。

| この引き数は、UNIX および Windows サーバーで指定できます。

-d *debug_level*

デーモン・サービスのデバッグ・レベルを、指定した値に設定します。有効な値は 1、2、または 3 です。このオプションは、*install* および *start* アクションとともに使用できます。

| この引き数は、UNIX および Windows サーバーで指定できます。

-u *user_id*

指定したユーザー ID で実行するように、デーモン・サービスを設定します。このオプションは、*install* アクションとともに使用できます。

この引き数は、Windows サーバーでのみ指定できます。

-p *password*

指定したユーザー ID のパスワードを指定します。パスワードが有効かつ必要なのは、**-u** オプションを指定する場合だけです。**-u** オプションの設定時に **-p** オプションを指定しなかった場合、パスワードを求めるプロンプトがプログラムによって出されます。このオプションは、*install* アクションとともに使用できます。

この引き数は、Windows サーバーでのみ指定できます。

start アクションで指定されるオプションは、現在のデーモンの実行のみに影響し、*install* アクションを使用して指定される値をオーバーライドします。

例:

以下の例は、デーモン・アクションを示しています。これらの例では、BLAST_DAEMON.config ファイルが *db2blast_daemon.exe* と同じディレクトリ内にあることを想定しています。

- デーモンの状況をチェックするには、次のようにします。

```
db2blast_daemon -a status
```

- デバッグをオンにした状態でデーモン・サービスをインストールするには、次のようにします。

```
db2blast_daemon -a install -d 2
```

- デーモンを開始するには、次のようにします。

```
db2blast_daemon -a start
```

- デーモンを停止するには、次のようにします。

```
db2blast_daemon -a stop
```

- デーモン・サービスを除去するには、次のようにします。

```
db2blast_daemon -a remove
```

関連タスク:

- 117 ページの『BLAST デーモンの開始』

BLAST ラッパーの登録

BLAST ラッパーの登録は、BLAST データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

BLAST データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上にラッパーを blast_wrapper の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER blast_wrapper LIBRARY 'libdb2lsblast.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『BLAST ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『BLAST ラッパーのサーバー定義の登録』です。

関連資料:

- 120 ページの『BLAST ラッパーのライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

BLAST ラッパーのライブラリー・ファイル

次の表では、BLAST ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、libdb2lsblast.a、libdb2lsblastF.a、および libdb2lsblastU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 24. BLAST ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパーのライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lsblast.a
Linux	/opt/IBM/db2/V8.1/lib	libdb2lsblast.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lsblast.so
Windows	%DB2PATH%¥bin	db2lsblast.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:¥Program Files¥IBM¥SQLLIB です。

関連タスク:

- 119 ページの『BLAST ラッパーの登録』

BLAST データ・ソースのサーバーの登録

BLAST データ・ソースのサーバーの登録は、BLAST をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

BLAST サーバーをフェデレーテッド・システムに登録するには、CREATE SERVER ステートメントを使用します。

ご使用の環境において、BLAST 可能ファイルとデーモンがインストールされている各マシンについて、その BLAST 実行可能ファイルとデーモンのインスタンスを使用して実行したい BLAST 検索のタイプごとに 1 つのサーバーを登録する必要があります。

たとえば、BLASTn 検索に使用される CREATE WRAPPER ステートメントを使用して作成した my_blast ラッパーの blast_server1 というサーバーを登録するには、以下のステートメントをサブミットします。

```
CREATE SERVER blast_server1
  TYPE blastn
  VERSION 2.1.2
  WRAPPER my_blast
  OPTIONS (NODE 'big_rs.company.com', DAEMON_PORT '4007')
```

この一連の作業における次のタスクは、『BLAST データ・ソースのニックネームの登録』です。

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 590 ページの『CREATE SERVER ステートメントの引き数 - BLAST ラッパー』

BLAST データ・ソースのニックネームの登録

BLAST データ・ソースのニックネームの登録

BLAST データ・ソースのニックネームの登録は、BLAST をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で BLAST データ・ソースを参照する場合に使用されます。

手順:

BLAST ニックネームを登録するには、CREATE NICKNAME ステートメントを使用します。

BLAST 検索の各タイプは別々のサーバーで取り扱われるので、所定の BLAST 可能データ・ソースに対して実行したい BLAST 検索の各タイプごとに別々のニックネームを定義する必要があります。

ニックネームを作成する際に、データ・ソースの定義行部分の列情報を指定します。他のすべての列は固定されています。

一連の作業はこれで終わりです。

関連概念:

- 122 ページの『定義行の構文解析』

関連タスク:

- 111 ページの『フェデレーテッド・サーバーへの BLAST データ・ソースの追加』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 591 ページの『CREATE NICKNAME ステートメント構文 - BLAST ラッパー』
- 127 ページの『CREATE NICKNAME ステートメント - BLAST ラッパーの例』
- 123 ページの『BLAST ニックネームの固定列』

定義行の構文解析

定義行は、BLAST 可能データ・ソースの各シーケンスに対するキーのようなもので、各 BLAST ヒットの一部として戻されます。定義行は *define* とも呼ばれます。

定義行に関して BLAST ラッパーによって戻され、構文分析される値は、オリジナルの FASTA ファイル内の定義行と同一であるとは限りません。たとえば、BLAST ヒットの Accession 番号フィールドにデータがある場合、戻される定義行には、Accession 番号データとそれに続く Definition フィールド・データが含まれます。そして、戻されるデータをラッパーが構文解析します。

推奨事項: ラッパーが定義行が戻したり構文解析する方法を判別するために、1 列の定義行があるニックネームを作成してください。次いで、照会を実行し、特定のデータ・ソースに関する定義行のラッパーによって戻される形式を参照してください。

定義行情報を自分の結果表に含めるには、CREATE NICKNAME ステートメントに定義行の列を指定する必要があります。指定する列ごとに、INDEX オプションおよび DELIMITER オプションを含めなければなりません。指定する最後の列に定義行情報の残りを含める場合、その最後の列で DELIMITER オプションを省略できます。

定義行の列の有効なデータ・タイプは CLOB、DOUBLE、FLOAT、INTEGER、および VARCHAR です。

関連概念:

- 506 ページの『定義行構文解析のユーザー定義関数 - 概要』

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 591 ページの『CREATE NICKNAME ステートメント構文 - BLAST ラッパー』
- 127 ページの『CREATE NICKNAME ステートメント - BLAST ラッパーの例』
- 123 ページの『BLAST ニックネームの固定列』

BLAST ニックネームの固定列

BLAST データ・ソースに対して CREATE NICKNAME ステートメントを発行すると、そのニックネームで固定列のセットが自動的に作成されます。

固定列はニックネームに関する定義の一部であり、フェデレーテッド・データベース・システムのカatalog内作成されます。SQL 照会で固定列を参照できます。固定列には入力固定列と出力固定列の 2 つのタイプがあります。

BLAST ニックネームの固定入力列: 固定入力列は WHERE 文節で指定されます。入力列は、SQL 照会のパラメーター受け渡し述部として使用されます。それら固定列は、標準の BLAST スイッチを BLAST に渡します。次に、これらのスイッチを使用して、指定されたデータ・ソースに対して BLAST が実行されます。固定入力列は、照会の SELECT リストの中で参照することもでき、結果表の一部として戻されます。

以下の表は、WHERE 文節で使用できる固定列をリストしています。

表 25. BLAST ニックネームの固定入力列

名前	データ・タイプ	演算子	説明
BlastSeq	VARCHAR (32000) または CLOB	=	照会シーケンスを BLAST ラッパーに渡します。
E_Value	DOUBLE	<	入力および出力パラメーターの両方。入力パラメーターとしては、この列は、blastall から戻されるべき期待値の上限を BLAST ラッパーに示します。
QueryStrands	INTEGER	=	BLASTn 検索を実行するときに比較されるべき成分を指定します。1 の値は、トップの成分を使用すべきことを示します。2 は、ボトム成分を示し、そして 3 は両方の成分が比較されるべきことを示します。
GapAlign	CHAR(1)	=	BLAST 出力でギャップのある配列が許されるかどうかを、ラッパーに指示します。

表 25. BLAST ニックネームの固定入力列 (続き)

名前	データ・タイプ	演算子	説明
Matrix	VARCHAR(50)	=	アミノ酸のペアの間での類似性の程度を決定するために、blastall によって使用される置換マトリックスを決定します。アミノ酸をアミノ酸と比較する BLAST 検索タイプのみが、この述部を使用します。
NMisMatchPenalty	INTEGER	=	同種領域の中のヌクレオチドのペアの1 つが一致しない場合に、配列のスコアから blastall が差し引く値を指定します。ヌクレオチドをヌクレオチドと比較する BLAST 検索タイプのみが、この述部を使用します。
NMatchReward	INTEGER	=	同種領域の中の一致するヌクレオチドのペアのそれぞれに対して、配列のスコアに blastall が加える値を指定します。ヌクレオチドをヌクレオチドと比較する BLAST 検索タイプのみが、この述部を使用します。
FilterSequence	CHAR(1)	=	生物学的に関心を引かないセグメントを、照会シーケンスから除去するためのフィルター操作を実行するかどうかを、blastall に指示します。検索タイプが BLASTn の場合、使用するフィルターは DUST です。それ以外の場合、フィルター操作は SEG によって実行されます。
NumberOfAlignments	INTEGER	=	BLAST 出力に含める HSP 配列の数を指定します。
GapCost	INTEGER	=	配列の長さを伸ばすことを可能にするために、照会シーケンスまたはヒット・シーケンスのどちらかにギャップを導入する必要がある場合に、配列のスコアから blastall が差し引く値を指定します。
ExtendedGapCost	INTEGER	=	配列の長さを伸ばすことを可能にするために、照会シーケンスまたはヒット・シーケンスのどちらかにすでに導入されているギャップを、ヌクレオチドまたはアミノ酸の 1 つ分だけ拡張する必要がある場合に、配列のスコアから blastall が差し引く値を指定します。
WordSize	INTEGER	=	blastall が最初にデータベースで検索する初期ヒットの長さを blastall に示します。

表 25. BLAST ニックネームの固定入力列 (続き)

名前	データ・タイプ	演算子	説明
ThresholdEx	INTEGER	=	BLAST が、そのしきい値より下へはそれ以上ヒットを拡張することをしないというスコアのしきい値を示します。

ニックネームを作成する場合、列に関するデフォルトのデータ・タイプをオーバーライドできます。たとえば、HSP_H_Seq や HSP_Midline 列など、いくつかの列は大量のデータを戻す可能性があります。列の最初の 50 バイトを戻すために、データ・タイプ VARCHAR(50) を指定して、列を定義することができます。最初の 50 バイトだけが出力列にコピーされます。

固定入力列の BLAST 検索タイプおよびスイッチ: 以下の表では、各固定入力列に対するサポートされている BLAST 検索タイプおよびスイッチがリストされています。

表 26. 入力固定列によってサポートされている BLAST 検索タイプおよびスイッチ

名前	BLAST 検索タイプ	BLAST スイッチ	必須	デフォルト
BlastSeq	n、p、x、tn、tx	-l	はい	N/A
E_Value	n、p、x、tn、tx	-e	いいえ	10
QueryStrands	n	S	いいえ	3
GapAlign	n、p、x、tn、tx	-g	いいえ	T
Matrix	p、x、tn、tx	-n	いいえ	BLOSUM62
NMisMatchPenalty	n	-q	いいえ	-3
NMatchReward	n	-r	いいえ	1
FilterSequence	n、p、x、tn、tx	-F	いいえ	T
NumberOfAlignments	n、p、x、tn、tx	-b	いいえ	250
GapCost	n、p、x、tn、tx	-G	いいえ	11
ExtendedGapCost	n、p、x、tn、tx	-E	いいえ	1
WordSize (Blastn の場合、7 より小さい値は無効)	n、p、x、tn、tx	-W	いいえ	11 -BLASTn 3 -BLASTp
ThresholdEx	n、p、x、tn、tx	-f	いいえ	0

BLAST ニックネームの固定出力列: 以下の表は、WHERE 文節で使用できる固定列をリストしています。

表 27. BLAST ニックネームの固定出力列

名前	データ・タイプ	説明
Score	DOUBLE	BLAST 結果に報告されている HSP の計算スコア。

表 27. BLAST ニックネームの固定出力列 (続き)

名前	データ・タイプ	説明
E_value	DOUBLE	入力および出力パラメーターの両方。出力パラメーターとして、この列は BLAST 結果に報告されている HSP の計算スコアを提供します。
Length	INTEGER	BLAST 結果に報告されるヒット・シーケンスの長さ。
HIT_NUM	INTEGER	1 から始まる、BLAST 結果に報告されるヒット番号
HSP_NUM	INTEGER	1 から始まる、BLAST 結果に報告される HSP 番号
HSP_Info	VARCHAR(100)	BLAST によって報告される、与えられた HSP の情報ストリング。このストリングには、照会シーケンスとヒット・シーケンスの間で一致した、ヌクレオチドまたはアミノ酸の数についての情報が含まれています。
HSP_ALIGNMENT_LENGTH	INTEGER	HSP 配置の長さ。
HSP_IDENTITY	INTEGER	配置の長さで分割された ID の数として定義される、配置のパーセント ID。
HSP_GAPS	INTEGER	配置の長さで分割されたギャップの数として定義される、配置でのパーセント・ギャップ。
HSP_POSITIVE	INTEGER	配置の長さで分割された正数の数として定義される、配置のパーセント正数。
HSP_QUERY_FRAME	INTEGER	照会シーケンスでの配置の読み取りフレーム。 blastx、tblastn、および tblastx タイプのサーバーにのみ使用できます。
HSP_HIT_FRAME	INTEGER	ヒット・シーケンス内の配置の読み取りフレーム。 blastx、tblastn、および tblastx タイプのサーバーにのみ使用できます。
HSP_Q_Start	INTEGER	照会シーケンス上の最初の同種ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_Q_End	INTEGER	照会シーケンス上の最後の同種ヌクレオチドまたはアミノ酸の数字で表した位置。

表 27. BLAST ニックネームの固定出力列 (続き)

名前	データ・タイプ	説明
HSP_Q_Seq	VARCHAR(32000)	HSP_Q_Start で始まり HSP_Q_End で終わる照会シーケンスのセグメント。 この列に関するデフォルトのデータ・タイプをオーバーライドし、最大長 5 メガバイトで CLOB を指定できません。
HSP_H_Start	INTEGER	ヒット・シーケンス上の最初の同種ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_H_End	INTEGER	ヒット・シーケンス上の最後の同種ヌクレオチドまたはアミノ酸の数字で表した位置。
HSP_H_Seq	VARCHAR(32000)	HSP_H_Start で始まり HSP_H_End で終わるヒット・シーケンスのセグメント。 この列に関するデフォルトのデータ・タイプをオーバーライドし、最大長 5 メガバイトで CLOB を指定できません。
HSP_Midline	VARCHAR(32000)	照会シーケンスおよびヒット・シーケンスの同種領域における各位置でのアミノ酸間またはヌクレオチド間での同種関係の程度を示す BLAST によるストリング出力。 この列に関するデフォルトのデータ・タイプをオーバーライドし、最大長 5 メガバイトで CLOB を指定できません。

関連概念:

- 122 ページの『定義行の構文解析』

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 591 ページの『CREATE NICKNAME ステートメント構文 - BLAST ラッパー』
- 127 ページの『CREATE NICKNAME ステートメント - BLAST ラッパーの例』

CREATE NICKNAME ステートメント - BLAST ラッパーの例

以下の CREATE NICKNAME ステートメントは、ニックネーム genbank を定義します。

ここでは、BLAST 結果の定義フィールドに、以下の情報が入っているものと想定しています。

```
>276342 15:8924 PMON5426
```

ここで、

276342

BLAST 結果の Accession フィールド。

15:8924 PMON5426

有機体番号、それに続く実験番号そしてその次に固有 ID が含まれている、BLAST 結果の中の定義フィールド。

この情報により、以下のニックネームが作成されます。

```
CREATE NICKNAME genbank (  
  acc_num integer  OPTIONS(INDEX '1', DELIMITER ' '),  
  org_num integer  OPTIONS(INDEX '2', DELIMITER ':'),  
  exp_num integer  OPTIONS(INDEX '3', DELIMITER ' '),  
  u_id varchar(10) OPTIONS(INDEX '4')  
  FOR SERVER blast_server1  
  OPTIONS(DATASOURCE 'genbank', TIMEOUT '300');
```

列 acc_num には 276342、列 org_num には 15、列 exp_num には 8924、そして列 u_id には PMON5426 が、それぞれ入ることになります。

CREATE NICKNAME ステートメントをサブミットした後、ニックネーム genbank を使用してフェデレーテッド・システムで照会することができます。フェデレーテッド・システムで、genbank ニックネームを他のニックネームや表と結合することもできます。

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 591 ページの『CREATE NICKNAME ステートメント構文 - BLAST ラッパー』

BLAST ラッパーで作業するための TurboBlast のセットアップ

制約事項:

TurboBlast は、特定の blastall コマンド・オプションをサポートしていません。たとえば、ギャップ付き配置オプション -g F はサポートされていません。BLAST ニックネームで GapAlign 列の値に F を指定すると、TurboBlast はエラーを生成します。サポートされないオプションの完全なリストは、「*TurboBlast 2.0 User Guide*」を参照してください。

手順:

BLAST ラッパーで作業できるように TurboBlast をセットアップするには、以下のようになります。

1. BLAST ラッパーをインストールして構成します。ブラスト可能データベースで照会を実行して、セットアップをテストします。
2. BLAST ラッパーおよび TurboBlast は、AIX、Linux、Solaris および Windows オペレーティング・システムをサポートしています。ただし、BLAST デーモンは、Windows オペレーティング・システムでは使用できません。BLAST デーモンが Windows オペレーティング・システム上にある場合、このデーモンはこれらのオペレーティング・システム上の TurboBlast で作業します。
3. 「*TurboBlast 2.0 Installation and Reference Guide*」に従って、TurboBlast をインストールして構成します。 TurboBlast システムはさまざまな方法で、インストールしてセットアップできます。 BLAST ラッパーが TurboBlast で作業できるようにするには、BLAST デーモンのあるコンピューター上に TurboBlast Client をインストールしてセットアップすることが必要です。 BLAST デーモンは、`tblastall` コマンドを呼び出すことができます。
4. TurboBlast をインストールして構成した後に、TurboBlast システムを必ずテストしてください。「*TurboBlast 2.0 Installation and Reference Guide*」の指示に従ってください。
5. `BLAST_DAEMON.config` ファイルを、次のように変更します。
 - a. `BLASTALL_PATH` パラメーターに、`tblastall` の絶対パスを指定します。例:
`BLASTALL_PATH=/home/blasttst/turboblast/TBlast-2.1/tblastall`。
 - b. TurboBlast にブラスト可能データベースをアップロードするのに使用したブラスト可能データベース名として、ブラスト可能データベース仕様項目を指定します。これらのデータベース名は、TurboBlast の下で `listdatabase -l` コマンドを入力すると表示されます。ブラスト可能データ・ソースへのパスではなく、この TurboBlast データベース名を使用してください。例:
`genbank=<the genbank database name in TurboBlast>`
6. BLAST デーモンを再始動します。BLAST デーモンは、`blastall` ではなく `tblastall` を呼び出し、ブラスト可能データベースでの検索作業を実行します。
7. `tblastall` に関連したログ・ファイルは、`BLAST_DAEMON.config` ファイルで指定された `DAEMON_LOGFILE_DIR` に書き込まれます。さらに、同じディレクトリーの BLAST デーモンで生成された `STDERR.log` および `STDOUT.log` もチェックしてください。

関連タスク:

- 111 ページの『フェデレーテッド・サーバーへの BLAST データ・ソースの追加』
- 113 ページの『BLAST デーモンの構成』

BLAST SQL 照会の作成

入力列の述部は、標準の BLAST スイッチを `blastall` 実行可能ファイルに渡すために使用されます。出力列の述部は、フェデレーテッド・サーバーによって処理されます。

制約事項:

有効であるためには、BLAST ラッパーへ渡されるすべての照会は、少なくとも BlastSeq 入力述部を含んでいなければなりません。他のすべての述部はオプションです。

手順:

BLAST 照会を作成するには、SQL ステートメントの WHERE 文節に入力述部を使用します。

以下の例は、3 つの入力述部 BlastSeq、GapCost、および NMisMatchPenalty を示しています。

```
Select * from blast b where  
BlastSeq = 'GTCCAGCC...' AND  
GapCost = -10 AND  
NMisMatchPenalty = -4;
```

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 130 ページの『BLAST データ・ソース - 照会の例』

BLAST データ・ソース - 照会の例

BLAST データ・ソースに対して照会がどのように作成されるかを示すために、いくつかのサンプルの BLAST 照会を示します。

照会を実行するために、ガイドとして例を使用します。

これらの照会において、各ニックネームに使用される名前は、BLAST 検索のタイプとデータ・ソースを示しています。サンプル照会ごとに登録ステートメントをリストする必要をなくすために、そのようにしています。また、照会の中には他の仮想のデータ・ソースを使用しているものがあります。これは、これらのサンプルが、他のデータ・ソースと結合されたときのラッパーの動作を示すことができるようにするためです。

照会 1

```
select *  
from blastn_genbank  
where BlastSeq =  
'caaccctccagccgagttgtcaatggcgaggaagctgttccccac';
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して、GenBank の BLASTn 検索を実行します。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 2

```
select *  
from blastn_genbank  
where BlastSeq =  
'caaccctccagccgagttgtcaatggcgaggaagctgttccccac'  
and GapCost = 8 and NmisMatchPenalty = -4;
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して、GenBank の BLASTn 検索を実行します。さらに、ラッパーは指示されている 2 つのパラメーターをデーモンへ渡します。そして、これらのパラメーターは blastall コマンド行へ渡されます。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 3

```
select blp.*
from blastp_swissprot blp, protein_db prdb
where prdb.keyword = 'malic enzyme'
and blp.BlastSeq = prdb.sequence;
```

この SQL ステートメントが実行されると、ラッパーは、仮想タンパク質シーケンス・データベースから戻されるシーケンスの数によって、SWISS-PROT の 0 または 1 回以上の BLASTp 検索を実行します。このステートメントは、DB2 によって 2 つの別々の照会に分割されます。1 つの BLASTp 検索は、仮想タンパク質データベースから戻される各行に対して実行されます。ラッパーは、入力パラメーター列および BLAST 結果列の両方を含むすべての選択可能な列を戻します。

照会 4

```
select Score, E_Value, HSP_Info, HSP_Q_Seq, HSP_H_Seq, HSP_Midline
from blastx_swissprot
where BlastSeq = 'gagttgtcaatggcgagg'
and GapCost = 8;
```

この SQL ステートメントが実行されると、ラッパーは、指示されているシーケンスを使用して SWISS-PROT の BLASTx 検索を実行します。この場合、blastall は、6 つのすべての読み取りフレームの中の入力シーケンスを変換し、6 つの新しく作成されたタンパク質シーケンスのそれぞれを使用して、同種関係検索を実行します。結果の HSP には、アミノ酸 - アミノ酸の配列が含まれます。ヌクレオチド - ヌクレオチドの配列ではありません。提供されたパラメーターはデーモンへ渡され、次にコマンド行経由で blastall へ渡されます。ラッパーは、照会の中で特定の要求されている列だけを戻します。

照会 5

```
select tblx.Score, tblx.E_Value, tblx.HSP_Info tblx.HSP_Q_Seq,
HSP_H_Seq, HSP_Midline
from tblastx_genbank tblx, gen_exp_database gedb
where tblx.BlastSeq = gedb.sequence
and gedb.organism = 'interesting organism'
and GapCost = 8
and FilterSequence = 'F';
```

この SQL ステートメントが実行されると、ラッパーは、仮想遺伝子表示データベースから戻されるシーケンスの数によって、GenBank の 0 または 1 回以上の tBLASTx 検索を実行します。このステートメントは、DB2 によって 2 つの別々の照会に分割されます。1 つの tBLASTx 検索は、仮想遺伝子表示データベースから戻される各行に対して実行されます。この場合、blastall は、入力シーケンスおよび 6 つのすべての読み取りフレームの中の GenBank のすべてのシーケンスを変換し、6 つの新しく作成された照会タンパク質シーケンスのそれぞれと、新しく作成されたすべてのデータベースタンパク質シーケンスを使用して同種関係検索を実行します。結果の HSP には、アミノ酸 - アミノ酸の配列が含まれます。ヌクレオチド -

ヌクレオチドの配列ではありません。提供されたパラメーターはデーモンへ渡され、次にコマンド行経由で `blastall` へ渡されます。ラッパーは、照会の中で特定の要求されている列だけを戻します。

関連資料:

- 209 ページの『Documentum データ・ソース - 照会の例』
- 248 ページの『Excel データ・ソース - 照会の例』

BLAST ラッパーの最適化のヒント

ネットワーク通信のパフォーマンスを改善するには、フェデレーテッド・サーバーと BLAST サーバーとを別のハードウェア上に置く必要があります。BLAST デーモンは、BLAST サーバー上に置いてください。

関連タスク:

- 113 ページの『BLAST デーモンの構成』

BLAST ラッパーのメッセージ

このセクションでは、BLAST のラッパーを処理するときに出されるメッセージをリストして説明します。

表 28. BLAST のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「sqlno_crule_save_plans [100]:rc (-2144272209) 空のプラン・リストが見つかった」。)	DB2 にサブミットされた SQL 照会はラッパーによって処理できませんでした。構文を訂正して再サブミットしてください。
SQL1816N	ラッパー "BLAST_WRAPPER" は、フェデレーテッド・データベースに定義しようとしているデータ・ソース (" <code><server type></code> ") の "type" にアクセスするためには使用できません。	CREATE SERVER ステートメントが無効な TYPE を使用していました。タイプは、サポートされている BLAST タイプの中のいずれか 1 つでなければなりません。
SQL1817N	CREATE SERVER ステートメントは、フェデレーテッド・データベースに定義したいデータ・ソースの "version" を示していません。	CREATE SERVER ステートメントがバージョンを指定していませんでした。

表 28. BLAST のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「デーモンに接続不可」です。	blast ラッパーはデーモンに接続できませんでした。デーモンが実行されていなかった可能性があります。構成が間違っている可能性があります。デーモンが実行されているマシンに到達できなかった可能性があります。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「Blast デーモンのタイムアウト」です。	CREATE NICKNAME ステートメントに指定されているタイムアウトが経過する前に、デーモンから何の結果も受け取りませんでした。タイムアウトを長くするか、デーモンに問題があるかどうかをチェックしてください。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「Blast Daemon が失敗」です。	デーモンが通信を停止したか、戻された結果が正しくフォーマットされていませんでした。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「blast デーモンからの不明エラー」です。	blast ラッパーが、認識できないエラー・コードをデーモンから受け取りました。デーモンのバージョンがラッパーのバージョンと互換性がなかった可能性があります。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「列の名前変更は不可」です。	列の中の 1 つの名前を変更しようとして、ALTER NICKNAME ステートメントが出されました。列の名前変更は許可されていません。
SQL1822N	データ・ソース "Blast Wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「XML パーサーのエラー」です。	Xerces パーサーが無効な状態にあるか、または例外をスローしました。
SQL1823N	サーバー "<server name>" からのデータ・タイプ "<data type name>" のデータ・タイプ・マッピングがありません。	指定したデータ・タイプはこの列ではサポートされません。
SQL1881N	"DEFAULT" は "<column-name>" に対して有効な "COLUMN" オプションではありません。	DEFAULT オプションが、それをサポートしない列で使用されていました。出力専用の列および定義行の列はデフォルト値をもちません。

表 28. BLAST のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"COLUMN" オプション "DEFAULT" は、 "<column-name>" に対して "<option-value>" に設定できません。	DEFAULT オプションに指定された値が、列に対して非互換のタイプの値であるか、間違ってフォーマットされています。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 8 章 ビジネス・アプリケーション・データ・ソースへのアクセスの構成

このセクションでは、WebSphere Business Integration ラッパーを使用してビジネス・アプリケーション・データ・ソースをフェデレーテッド・システムに追加する方法を説明します。

WebSphere Business Integration ラッパー

WebSphere[®] Business Integration ラッパーは読み取り専用ラッパーで、WebSphere Business Integration アダプターを使用してビジネス・アプリケーションにアクセスします。サポートされているアダプターとアプリケーションについては、「*IBM DB2 Information Integrator フェデレーテッド・システム・ガイド*」を参照してください。

WebSphere Business Integration ラッパーは、SAP、Siebel、および PeopleSoft によって作成されたものを含めた、ビジネス・アプリケーションへの SQL インターフェースを提供します。WebSphere Business Integration ラッパーを使用することにより、フェデレーテッド・システム関数を使用して、ビジネス・アプリケーションからのビジネス・データを他のフェデレーテッド・データ・ソース上のデータと結合させることができます。WebSphere Business Integration ラッパーは、ビジネス・オブジェクト定義を抽出してニックネームの階層に展開します。

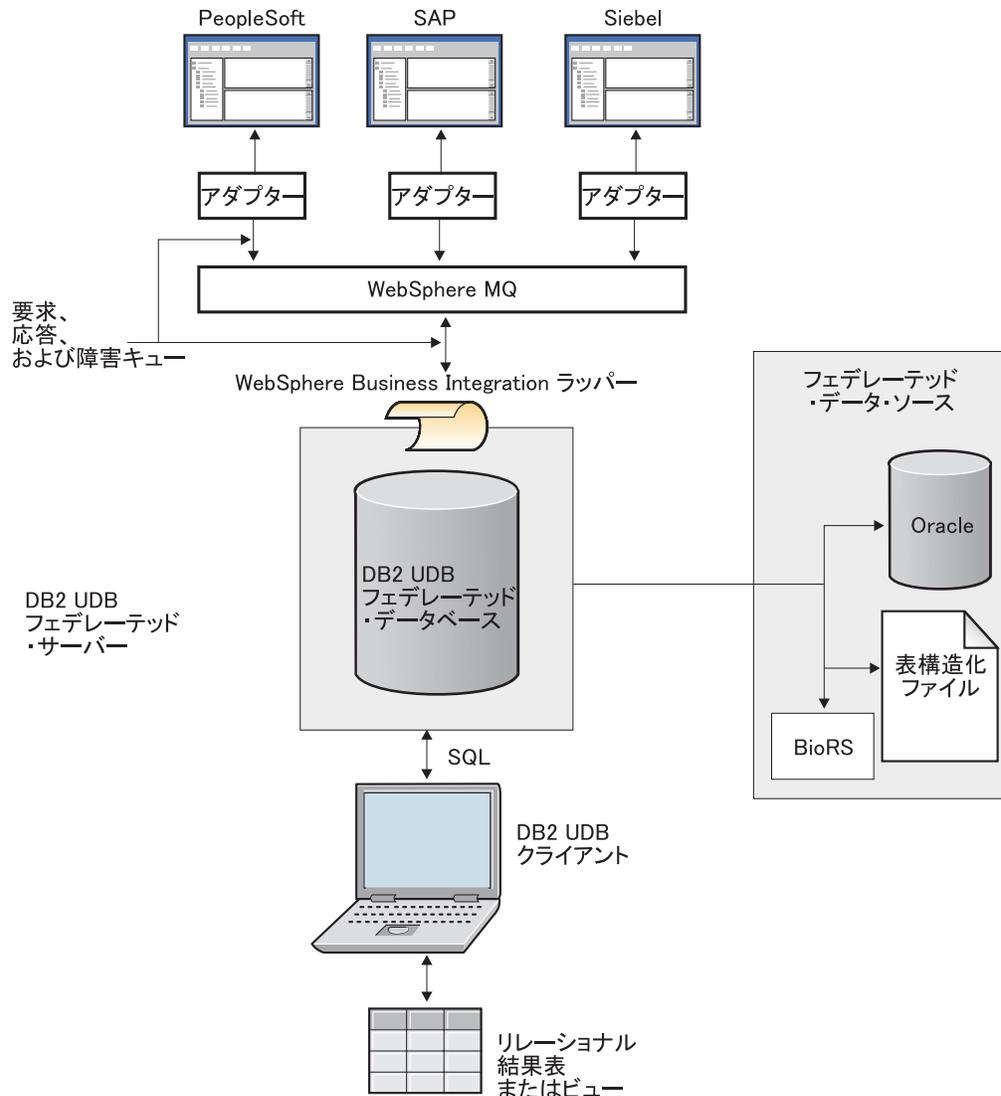


図9. DB2[®] Universal Database 環境での WebSphere Business Integration ラッパー

図9は、DB2 Universal Database[™] 環境における、WebSphere Business Integration ラッパーとアダプターの関係を示しています。以下のステップには、フェデレーテッド・システム内のビジネス・アプリケーション・データにアクセスするプロセスが説明されています。

1. ユーザーがフェデレーテッド・サーバーに照会を送信して、Siebel アプリケーションなどのデータ・ソースにマップするニックネームにアクセスする。
2. ラッパーが照会をビジネス・オブジェクトにトランスフォームする。
3. ラッパーがこのビジネス・オブジェクトを WebSphere MQ メッセージ・キューに置く。
4. アプリケーションの WebSphere Business Integration アダプターが、メッセージ・キューからビジネス・オブジェクト (これが要求となる) を読み取る。
5. WebSphere Business Integration アダプターがビジネス・アプリケーションと連動して、応答ビジネス・オブジェクトを準備する。

6. WebSphere Business Integration アダプターが応答ビジネス・オブジェクトをメッセージ・キューに書き込む。
7. ラッパーが応答キューから応答ビジネス・オブジェクトを読み取る。
8. ラッパーは応答業務・オブジェクトを抜き出し、ニックネーム定義に定義されているリレーショナル・スキーマに基づいて結果セットに展開する。

関連概念:

- 137 ページの『ビジネス・オブジェクト定義』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 139 ページの『WebSphere Business Integration アダプターの構成』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』

ビジネス・オブジェクト定義

ビジネス・オブジェクト定義は、WebSphere® Business Integration アダプターがビジネス・オブジェクトのインスタンスを作成するときの元になるテンプレートです。ビジネス・オブジェクト定義とはビジネス・アプリケーション・データ・エンティティのことです。

ビジネス・オブジェクト定義はビジネス・オブジェクト定義のインスタンスであり、フラット構造または階層構造のどちらにすることもできます。フラット・ビジネス・オブジェクトには属性が 1 つだけ含まれます。階層ビジネス・オブジェクトには 1 つ以上の複合属性が含まれます。ビジネス・オブジェクト定義のリポジトリは、サポートされている特定のアプリケーション用に各 WebSphere Business Integration アダプター内に存在しています。

フラット・ビジネス・オブジェクトの例を以下に示します。

Customer

```
Gomez
Juanita
Apt 2C
123 Main Street
Big City
California
91234
888
1111111
```

複数の複合属性を持つ階層ビジネス・オブジェクトを以下に示します。

```

Contact (Parent)
  ID
  Customer ID
  Date
  Text
  Authorization
  Line items (there are 0 or more Line item elements)
    (Child elements)
    Business object 1
    Business object 2
    Business object 3

```

ビジネス・オブジェクト定義は、各 WebSphere Business Integration アダプターとともにパッケージされている Object Discovery Agent ツールを使用して生成することが必要です。Object Discovery Agent ツールは、ビジネス・オブジェクト定義の XML スキーマ定義ファイルを生成します。ビジネス・オブジェクトが階層構造として定義されている場合、Object Discovery Agent ツールは複数のスキーマ・ファイルを生成する場合があります。

XML スキーマ定義は、ファイル・タイプ .xsd のファイルで、WebSphere Business Integration 構成に指定されたディレクトリにあります。WebSphere Business Integration ラッパーのニックネームを作成する前に、ビジネス・オブジェクト定義を生成する必要があります。WebSphere Business Integration アダプターのファミリーの詳細については、www.ibm.com/websphere/integration/wbiadapters を参照してください。

ニックネームを作成するには、Object Discovery Agent ツールが作成する xsd ファイルを使用します。ニックネームは、ビジネス・オブジェクト定義のリレーショナル・スキーマ表記を提供します。WebSphere Business Integration ラッパーは、階層ビジネス・オブジェクトをリレーショナル・ニックネームにマップします。たとえば、カーディナリティー 'n' の子ビジネス・オブジェクトはそれぞれ、外部キー制約により、親ビジネス・オブジェクトのニックネームにリンクされている個別のニックネームにマップされます。

IBM® DB2® Information Integrator にアクセス可能な WebSphere Business Integration ビジネス・オブジェクトは、次の表に示されているように特定のアプリケーション・エンティティーにマップされます。

表 29. ビジネス・オブジェクトおよび関連付けられているアプリケーション・エンティティー

ビジネス・オブジェクト	アプリケーション・エンティティー
Siebel	ビジネス・コンポーネント
PeopleSoft	コンポーネント・インターフェース
SAP	BAPI

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 139 ページの『WebSphere Business Integration アダプターの構成』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』

WebSphere Business Integration アダプターの構成

フェデレーテッド・ラッパー関数を使用して SQL ステートメントからビジネス・アプリケーションにアクセスしようとする場合、WebSphere Business Integration アダプターをインストールして構成する必要があります。各アダプターはフェデレーテッド・サーバー定義にマップします。

前提条件:

- サポートされているアダプターについては、「*IBM DB2 Information Integrator* フェデレーテッド・システム・ガイド」を参照してください。
- 各アダプターのインストール情報については、IBM WebSphere Business Integration Information Center を参照してください。
- 特定のビジネス・アプリケーションに関するヘルプは、WebSphere Business Integration Adapter 資料を参照してください。
- 使用する特定のアダプターの最新フィックスパックすべてをインストールします。アダプター関連のサポート情報はWebSphere Business Integration Adapters Support サイトから入手できます。
- メッセージ・キューの構成に関する情報については、「*WebSphere MQ* システム管理ガイド」を参照してください。
- アダプターおよび構成プロパティについては、IBM WebSphere Business Integration Information Center でインストール情報を参照してください。

手順:

WebSphere Business Integration アダプターを構成するには、次のようにします。

1. Object Discovery Agent ツールと Business Object Designer ツールを構成し、ビジネス・オブジェクト定義を作成する。

Business Object Designer ツールでビジネス・オブジェクト定義を構成するときには、ビジネス・アプリケーションに合わせて以下の verb 値を指定します。

表 30. ビジネス・アプリケーションで使用される verb 値

ビジネス・アプリケーション	Verb
SAP	Retrieve
PeopleSoft	Retrieve
Siebel	RetrieveByContent

Object Discovery Agent ツールを構成および使用する方法の詳細については、構成するアダプターの文書を参照してください。

2. WebSphere Business Integration アダプター・インターフェースから Connector Configurator ツールを使用して、以下の情報を含む構成ファイルを定義する。
 - アダプターによってサポートされるビジネス・オブジェクト。
 - アダプターの構成プロパティ。標準構成プロパティとアプリケーション固有の構成プロパティがあります。

標準構成プロパティ

IBM DB2 Information Integrator でアダプターを使用するには、一部のプロパティ値をカスタマイズする必要があります。構成対象となるプロパティは以下のとおりです。

- Integration Broker の値を WMQI に指定する。
- アダプターが所有するメタデータ・リポジトリのロケーションを指定する。ビジネス・オブジェクト定義を含む XML スキーマ定義ファイルは、このロケーションに保管されます。
- デリバリー・トランスポートのタイプを WMQI-MQ に指定する。
- アダプターが使用するキューを管理するキュー・マネージャーの名前を指定する。
- アダプターを実行するために必要な 8 つのキューの名前を指定する。

アプリケーション固有の構成プロパティ

このプロパティは、アプリケーション固有のコンポーネントの値を指定します。指定した値は、アプリケーションとのセッションの確立に役立ちます。プロパティは、アプリケーション固有のコンポーネントの処理動作を指示します。

3. ラッパーに必要な 3 つの WebSphere MQ メッセージ・キュー、`request_queue`、`response_queue`、`fault_queue` を定義する。

WebSphere MQ は、アダプターとラッパーの間のメッセージングおよびトランスポートの層です。

request_queue

DB2 Information Integrator からアダプターに要求メッセージを配信します。

response_queue

アダプターから DB2 Information Integrator に応答メッセージを配信します。

fault_queue

アダプターから DB2 Information Integrator に障害メッセージを配信します。アダプターは、応答キューにメッセージを置けないときに、このキューにメッセージを置きます。

これらのキューは静的キューで、アダプターとラッパーとの間でメッセージ (データ・オブジェクトやエラー・メッセージを含む) を交換するために使用されません。

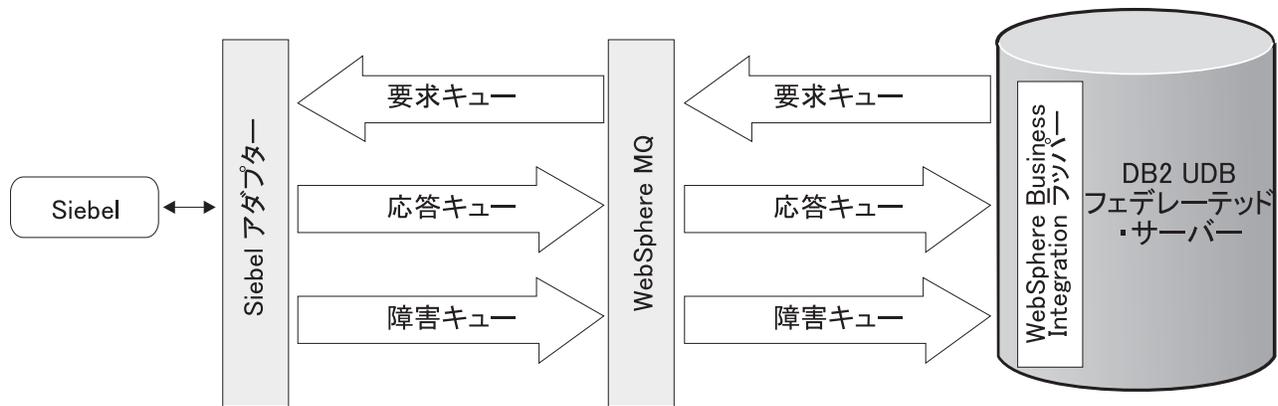


図 10. Siebel ビジネス・アプリケーションと DB2 フェデレーテッド・サーバーとの間で情報をトランスポートする WebSphere メッセージ・キューのトポロジー

4. アダプターに必要な以下の 5 つの追加メッセージ・キューを定義する。

- AdminInQueue
- AdminOutQueue
- SynchronousRequestQueue
- SynchronousResponseQueue
- DeliveryQueue

WebSphere Business Integration アダプターには、DB2 Information Integrator ではなく WMQI ブローカーでアダプターが使用されるときに使う、5 つの追加のキューが必要です。これらの追加メッセージ・キューを作成および構成すると、アダプターを開始することができます。

5. 以下のいずれかの方法で、WebSphere MQ ユーザー許可を定義する。

- DB2 インスタンス所有者 ID を MQManager グループの一部として定義する。
- MQManager 管理者に、ServerConnection チャンネルの作成時に MCAUSER 値を設定してもらう。MCAUSER の値は、MQManager グループまたは管理者グループに属するユーザー ID でなければなりません。

関連概念:

- 137 ページの『ビジネス・オブジェクト定義』
- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 143 ページの『WebSphere Business Integration ラッパーの登録』

関連資料:

- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』

フェデレーテッド・サーバーへのビジネス・アプリケーションの追加

ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加

フェデレーテッド・サーバーがビジネス・アプリケーション・データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供する必要があります。

ビジネス・アプリケーション・データ・ソースへのアクセスは、DB2 コントロール・センターまたは DB2 コマンド行を使用して構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーを構成するために必要なステップを順番に提供するウィザードが組み込まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- 32 ビット DB2 UDB インスタンスを使用するフェデレーテッド・データベースがフェデレーテッド・サーバー上になければなりません。

手順:

ビジネス・アプリケーション・データ・ソースをフェデレーテッド・システムに追加するには、次のようにします。

1. WebSphere Business Integration ラッパーを登録する。
2. サーバー定義を登録する。
3. ビジネス・アプリケーション・データ・ソースのニックネームを登録する。
4. オプション: WebSphere Business Integration ニックネームのフェデレーテッド・ビューを作成する。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』
- 137 ページの『ビジネス・オブジェクト定義』
- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール後の Oracle genclntsh スクリプトの編集および libclntsh ファイルの作成 (HP-UX, Linux, Solaris)』
- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 145 ページの『ビジネス・アプリケーション・データ・ソースのサーバー定義の登録』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

- 155 ページの『ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 57 ページの『フェデレーテッド・データベースの作成』
- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 145 ページの『ビジネス・アプリケーション・データ・ソースのサーバー定義の登録』
- 155 ページの『ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成』
- 139 ページの『WebSphere Business Integration アダプターの構成』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』

WebSphere Business Integration ラッパーの登録

WebSphere Business Integration ラッパーの登録は、フェデレーテッド・サーバーにビジネス・アプリケーション・データ・ソースを追加するという、より大きな作業の一部を成しています。

ビジネス・アプリケーション・データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。ラッパーは、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。DB2 コントロール・センターには、ラッパーを登録するために必要なステップを順番に提供するウィザードが組み込まれています。

前提条件:

1. 適切なアダプターをインストールして構成する
2. WebSphere MQ バージョン 5.3 (CSD レベル 5) をインストールして構成する。
3. WebSphere MQ メッセージ・キューを作成する。
4. WebSphere MQ マネージャーが DB2 Information Integrator と同じシステム上にない場合は、DB2 Information Integrator サーバー・インスタンスをインストールしたシステムに、WebSphere MQ バージョン 5.3 (CSD レベル 5) クライアントをインストールする必要があります。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

たとえば、Windows オペレーティング・システムを使用するフェデレーテッド・サーバーに wbi_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER wbi_wrapper LIBRARY 'db2wbi.dll';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『WebSphere Business Integration ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『WebSphere Business Integration ラッパーのサーバー定義の登録』です。

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』

関連資料:

- 144 ページの『WebSphere Business Integration ラッパー・ライブラリー・ファイル』
- 144 ページの『WebSphere Business Integration ラッパー・ライブラリー・ファイル』

WebSphere Business Integration ラッパー・ライブラリー・ファイル

次の表には、WebSphere Business Integration ラッパーのディレクトリー・パスおよびライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは libdb2wbi.a、libdb2wbiF.a、および libdb2wbiU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 31. WebSphere Business Integration ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2wbi.a
Windows	%DB2PATH%\%bin	db2wbi.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 143 ページの『WebSphere Business Integration ラッパーの登録』

ビジネス・アプリケーション・データ・ソースのサーバー定義の登録

ビジネス・オブジェクト・データ・ソースのサーバー定義の登録は、フェデレーテッド・システムにビジネス・オブジェクトを追加するという、より大きな作業の一部を成しています。

ラッパーを登録した後、対応するサーバーを登録する必要があります。

制約事項:

オプション `MQ_SVRCONN_CHANNELNAME` の指定は、オプション `MQ_CONN_NAME` を指定している場合にのみ行えます。オプション `MQ_CONN_NAME` のドロップは、オプション `MQ_SVRCONN_CHANNELNAME` をドロップするまで行えません。 `MQ_CONN_NAME` が指定されていない場合、フェデレーテッド・システムは `MQSERVER` 環境変数の値を使用します。 `MQSERVER` 環境変数は `db2dj.ini` ファイルに設定します。 `db2dj.ini` ファイルを編集する場合は、DB2 Universal Database を停止し、再起動する必要があります。

手順:

ビジネス・アプリケーションのサーバー定義をフェデレーテッド・システムに登録するには、`CREATE SERVER` ステートメントを発行します。

たとえば、Siebel ビジネス・アプリケーションのサーバー定義を登録するには、次のようにします。

```
CREATE SERVER siebel_server
  VERSION 2.4
  WRAPPER wbi_wrapper
  OPTIONS ( App_Type 'siebel',
           Request_Queue 'myqueue3',
           Response_Queue 'myqueue4',
           Fault_Queue 'myqueue5',
           MQ_Manager 'mymq'
           MQ_RESPONSE_TIMEOUT '55000',
           MQ_CONN_NAME '9.30.76.151(1420)',
           MQ_SVRCONN_CHANNELNAME 'SYSTEM.DEF.SVRCONN'
  )
```

この例の場合、ビジネス・アプリケーションは Siebel アプリケーションで、`APP_TYPE` オプションによって識別されています。有効値は `SIEBEL`、`PSOFT`、および `SAP` です。 `VERSION` オプションは、使用している WebSphere Business Integration アダプターのバージョンを表します。有効な値は 2.3 および 2.4 です。サーバー・オプションには、WebSphere Business Integration アダプターの構成に説明されているキュー定義が含まれている必要があります。 `MQ_RESPONSE_TIMEOUT` のデフォルト値は 50000 ミリ秒に設定されています。値 `-1` はタイムアウト制限なしを指定します。

この一連の作業における次のタスクは、『ビジネス・アプリケーション・データ・ソースのニックネームの登録』です。

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 139 ページの『WebSphere Business Integration アダプターの構成』
- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

関連資料:

- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』

ビジネス・アプリケーション・データ・ソースのニックネームの登録

ビジネス・アプリケーション・データ・ソースのニックネームの登録

ビジネス・アプリケーション・データ・ソースのニックネームの登録は、フェデレーテッド・システムにビジネス・アプリケーションを追加するという、より大きな作業の一部を成しています。

ニックネームは、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。DB2 コントロール・センターには、ニックネームを登録するために必要なステップを順番に提供するウィザードが組み込まれています。

ビジネス・オブジェクト定義の構造階層に対応したニックネームを作成する必要があります。親ニックネームには少なくとも 1 つの子ニックネームが含まれます。子ニックネームは、1 より大きいカーディナリティーを含み、かつ親ニックネームの元素内でネストされている元素に対応します。

手順:

ビジネス・アプリケーション・データ・ソースを DB2 コマンド行から登録するには、CREATE NICKNAME ステートメントを発行します。

たとえば、sieb_ssa_Contact_Contact という名前の Siebel ビジネス・オブジェクト定義のニックネームを登録するには、次のステートメントを発行します。

```
CREATE NICKNAME sieb_ssa_Contact_Contact_NN(  
  Id VARCHAR(15)  OPTIONS(XPATH './ns1:Id/text()'),  
    TEMPLATE '<ns1:Id>&column</ns1:Id>'),  
  FirstName VARCHAR(50)  OPTIONS(XPATH './ns1:FirstName/text()'),  
    TEMPLATE '<ns1:FirstName>&column</ns1:FirstName>'),  
  LastName VARCHAR(50)  OPTIONS(XPATH './ns1:LastName/text()'),  
    TEMPLATE '<ns1:LastName>&column</ns1:LastName>'),  
  AccountId VARCHAR(255)  OPTIONS(XPATH './ns1:AccountId/text()'),  
  PrimaryAccountName VARCHAR(100)  
    OPTIONS(XPATH './ns1:PrimaryAccountName/text()'),  
  PrimaryPostalCode VARCHAR(30)  
    OPTIONS(XPATH './ns1:PrimaryPostalCode/text()'),  
  PrimaryStreetAddress VARCHAR(200)
```

```

        OPTIONS(XPATH './ns1:PrimaryStreetAddress/text()'),
SalesRep VARCHAR(255) OPTIONS(XPATH './ns1:SalesRep/text()'),
State VARCHAR(255) OPTIONS(XPATH './ns1:State/text()')
FOR SERVER siebel_server
OPTIONS(XPATH '//ns1:sieb_ssa_contact_contact',
TEMPLATE '<ns1:sieb_ssa_contact_contact>
          &Id[1,1] &FirstName[0,1] &LastName[0,1]
        </ns1:sieb_ssa_contact_contact>',
BUSOBJ_NAME 'sieb_ssa_contact_contact',
NAMESPACES 'ns1="http://www.ibm.com/websphere/
            crossworlds/2002/BOSchema/
            sieb_ssa_contact_contact"');

```

BUSOBJ_NAME ニックネーム・オプションは、ビジネス・オブジェクト定義を表す XML スキーマ定義 (XSD) ファイルの名前です。

必須ステップ: XSD ファイル内の必須入力列にフラグを立てる

WebSphere Business Integration アダプターは、検索要求への応答として単一のビジネス・オブジェクトのみを戻すことができます。入力ビジネス・オブジェクトという形でのアダプターへの検索要求によってアプリケーション内内の複数のオブジェクトが識別される場合には、アダプターは最初のビジネス・オブジェクトのみを戻します。ラッパーは、完全な結果セットを検索できなかったことを示すエラーを出します。

検索要求への応答に単一のビジネス・オブジェクトだけが識別されるようにするには、要求ビジネス・オブジェクトのアダプターに対して十分な述部が与えられる必要があります。ラッパーからは、単一のビジネス・オブジェクトを識別するために必要な入力述部すべてを送信する必要があります。そのため、正しいテンプレート参照を使用することによって、列がニックネーム定義内で識別されなければなりません。以下のステップには、DB2 コントロール・センターを使用してニックネーム定義を生成する前に、正しい必須入力列を識別するために取るべき処置が説明されています。

1. SAP、Siebel、または PeopleSoft アプリケーション・リポジトリ内にあり、マップ対象のアプリケーション・エンティティのユニーク・キーを表す列を識別する。

SAP SAP Business Object Repository を使用して BAPI 用の必須入力パラメーターを識別することができます。BAPI は、WebSphere Business Integration Object Discovery Agent ツールにより、WebSphere Business Integration ビジネス・オブジェクトにマップされます。

Siebel 以下の方法のいずれかを使用します。

- Siebel アプリケーションには、各 Business Component に関連付けられた固有 ID 列があり、Business Component の各インスタンスごとにこの列に対して 16 進値を生成します。この ID 列は Business Component 階層の最上位に存在しており、生成された xsd ファイル内のエレメントにはすでに isKey="true" の指定でフラグが立っています (xml 注釈の appSpecificInfo セクション)。
- Siebel ツールを使用して、マップ対象の Business Component の複合ユニーク・キーを表すデータベース列を識別することができます。これらの列はすべて、ビジネス・オブジェクト階層の最上位またはルート・レベルにある必要があります。

PeopleSoft

Application Designer ツールを使用して、WebSphere Business Integration ビジネス・オブジェクト定義にマップされる階層の最上位レベル用の Component Interface 内で、getKey 列を識別します。

2. WebSphere Business Integration Object Discovery Agent ツールによってビジネス・オブジェクト定義用に生成された XSD ファイルを編集して、必須入力列にフラグを立てる。列にフラグを立てる場合のガイドラインは、ニックネームおよび列レベルでの TEMPLATE オプションのトピックにあります。
3. DB2 コントロール・センターからビジネス・オブジェクト定義用にニックネーム DDL を生成する。

ビジネス・アプリケーション・データ・ソースのニックネームを DB2 コントロール・センターから登録するには、次のようにします。

1. 「フェデレーテッド・データベース・オブジェクト」フォルダーを展開する。
2. ニックネームを登録するラッパー・フォルダーを展開する。
3. 「サーバー定義 (Server Definitions)」フォルダーを展開する。
4. ニックネームを登録するサーバー・フォルダーを展開する。
5. 「ニックネーム (Nicknames)」フォルダーを右マウス・ボタンでクリックして、「作成」を選択する。
6. 「ニックネームの作成 (Create Nicknames)」ウィンドウで「発見 (Discover)」をクリックして、データ・ソースでオブジェクトを選択するのに役立つ検索条件を定義する。
7. DB2 Information Integrator ユーザーのアクセス先にするビジネス・オブジェクトの定義を含む、XML スキーマ定義ファイルを指定する。
8. 「OK」をクリックして、選択した XML スキーマ定義ファイルに基づいてニックネームを作成する。

DB2 コントロール・センターは、スキーマ・ファイルを抽出して、適切な親子関係定義がある、複数のニックネーム作成 DDL ステートメントに展開します。作成されるニックネームは、XML スキーマ定義ファイルに定義されているビジネス・オブジェクト階層を表します。

オプション: この一連の作業における次のタスクは、『ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成』です。

関連概念:

- 149 ページの『ニックネームおよび列レベルでの TEMPLATE オプション』
- 137 ページの『ビジネス・オブジェクト定義』
- 135 ページの『WebSphere Business Integration ラッパー』
- 149 ページの『ニックネームおよび列レベルでの TEMPLATE オプション』
- 154 ページの『ニックネームと XPATH 式』

関連タスク:

- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』
- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』

- 143 ページの『WebSphere Business Integration ラッパーの登録』
- 155 ページの『ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ニックネームおよび列レベルでの TEMPLATE オプション

このトピックは、WebSphere® Business Integration ラッパーおよび Web サービス・ラッパーに適用されます。

WebSphere® Business Integration ラッパーと Web サービス・ラッパーは、WebSphere Business Integration アダプターと Web サービス環境に必要な XML 文書を作成します。これらのラッパーにはニックネーム作成時にニックネーム・レベルおよび列レベルのテンプレート・フラグメント (つまり、CREATE NICKNAME ステートメントの TEMPLATE オプション) が必要です。ラッパーは、照会計画中および照会実行段階で、この情報を使用します。

Web サービス・ラッパー

Web サービス・ラッパーの場合、必須およびオプションの属性は、WSDL 文書内の定義、および列の導出方法によって異なります。列は、エレメントまたはエレメントの属性のいずれから導出されます。

- 列がエレメントから導出されている場合、列がオプションかどうかは minOccurs 値によって判別されます。
- minOccurs の値が 0 である場合、列はオプションです。
- minOccurs の値が 1 である場合、列は必須です。
- 列がエレメントの属性から導出されている場合、列がオプションかどうかは属性に使用されている値により判別されます。
- 属性に値 use=optional が含まれている場合、列はオプションです。
- 属性に値 use=required が含まれている場合、列は必須です。

次の例は、列に関連付けられているスキーマ定義内の属性を示しています。

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="tns:ZooName"/>
    <xsd:element ref="tns:Count"/>
    <xsd:element ref="tns:LastModified"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:complexType>
```

WebSphere Business Integration ラッパー

WebSphere Business Integration ラッパーの場合、列が必須かオプションかは、アプリケーションおよび関連付けられているアダプターによって異なります。列に適したテンプレート・オプション値を指定することによって、必須およびオプションの入力列を識別する必要があります。DB2[®] コントロール・センターを使用してニックネームを作成する前に、XMLスキーマ定義ファイルを変更して、必須およびオプション入力列にフラグを立てなければなりません。

SAP BAPI

IBM[®] DB2 コントロール・センターは、XML スキーマ定義 (XSD) ファイル内にあるビジネス・オブジェクト定義を表す固有のフラグの値に基づいて、必須およびオプション入力列を判別します。

任意のレベルのビジネス・オブジェクト階層 (親または子ビジネス・オブジェクト) でエレメントの注釈セクションで、appSpecificInfo 値に I 接頭部があれば、ビジネス・オブジェクト定義のマップ先である SAP BAPI のインポート・パラメーターを表します。E 接頭部は、SAP BAPI のエクスポート・パラメーターを表します。一部のエレメントは、BAPI のインポートおよびエクスポートの両方のパラメーターとなることができます。次の例は、インポートおよびエクスポートの両方のパラメーターであるエレメントを示しています。

```
<bx:appSpecificInfo>ICOMPANYCODE:ECOMPANYCODE</bx:appSpecificInfo>
```

SAP ビジネス・オブジェクト・リポジトリから抽出された情報に基づいて、接頭部は WebSphere Business Integration Object Discovery Agent ツールにより自動的に生成されます。

インポート・パラメーター (appSpecificInfo 値に I 接頭部が指定されている) を表すエレメントに属性 minOccurs=1 が指定されている場合、DB2 コントロール・センターはそのエレメントを必須入力パラメーターとして識別し、エレメントを必須入力列としてニックネーム定義にフラグを立てます。WebSphere Business Integration Object Discovery Agent ツールは、SAP BAPI の必須入力パラメーターに対して自動的に minOccurs の値 1 を設定することはありません。ユーザー自身が SAP Business Object Repository を参照して、アクセスする BAPI 用の必須入力パラメーターすべてを判別する必要があります。次いで、XML スキーマ・ファイル内の対応するエレメントについて、手動で属性を minOccurs=1 に設定して編集する必要があります。入力パラメーターの minOccurs 属性値がデフォルト値である 0 のままであると、DB2 コントロール・センターは生成されるニックネーム階層に列をオプション入力列として指定します。

次の例は、オプション入力列を示しています。

```
<xsd:element name="Company_code" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICOMPANYCODE:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
```

```

        <xsd:maxLength value="4" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

次の例は、必須入力列を示しています。

```

<xsd:element name="Company_id" minOccurs="1">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICOMPANYID:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="true" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

SAP ビジネス・アプリケーション用の必須およびオプション入力列は、次の表に示されている構文によって指定されます。

表 32. SAP 入力列情報のフラグ設定スキーマ

SAP XSD ファイルで使用されるフラグ	必要入力列	ニックネーム・テンプレート内の列参照
接頭部 = 'I' および minOccurs=1 が指定された、階層の任意の場所にあるエレメント	はい	&columnname[1,1]
接頭部 = 'I' および minOccurs=0 が指定された、階層の任意の場所にあるエレメント	いいえ	&columnname[0,1]

Siebel および PeopleSoft

DB2 コントロール・センターは、エレメントの注釈の attributeInfo セクションにおける isRequired フラグの有無およびその値に基づいて、必須およびオプション入力列を判別します。isRequired フラグがない場合、その列は入力列ではありません。WebSphere Business Integration Object Discovery Agent ツールは、XSD ファイルにこれらのフラグを自動的に生成することはありません。ユーザー自身が必須およびオプション入力列を識別し、DB2 コントロール・センターを使用してニックネーム DDL を生成する前に、XSD ファイル内でそれらの列について適切にフラグを立てる必要があります。

次の例は、Siebel または PeopleSoft ビジネス・オブジェクト定義用の XSD ファイル内での必須入力列およびオプション入力列のフラグを示しています。

```

<xsd:element name="sieb_ssa_Contact_Contact">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version="1.0.0">
        <bx:appSpecificInfo>ON=Contact;CN=Contact</bx:appSpecificInfo>
      </bx:boDefinition>
    </xsd:appinfo>
  </xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Id" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>FN=Id</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false"
          isKey="true" isRequired="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
...

```

図 11. Siebel ビジネス・オブジェクト定義の一部分 (1/2)

```

...
<xsd:element name="FirstName" minOccurs="1">
<xsd:annotation>
  <xsd:appinfo>
    <bx:boAttribute>
      <bx:appSpecificInfo>FN=First Name</bx:appSpecificInfo>
      <bx:attributeInfo isForeignKey="false" isKey="false"
        isRequired="false" />
    </bx:boAttribute>
  </xsd:appinfo>
</xsd:annotation>
<xsd:simpleType>
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="50" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="LastName" minOccurs="1">
<xsd:annotation>
  <xsd:appinfo>
    <bx:boAttribute>
      <bx:appSpecificInfo>FN=Last Name</bx:appSpecificInfo>
      <bx:attributeInfo isForeignKey="false" isKey="false"
        isRequired="false" />
    </bx:boAttribute>
  </xsd:appinfo>
</xsd:annotation>
...

```

図 11. Siebel ビジネス・オブジェクト定義の一部分 (2/2)

Siebel および PeopleSoft ビジネス・アプリケーション用の必須およびオプション入力列は、次の表に示されている構文によって指定されます。

表 33. Siebel および PeopleSoft 入力列情報のフラグ設定スキーマ

Siebel および PeopleSoft XSD ファイルで使用されるフラグ	必要入力列	ニックネーム・テンプレート内の列参照
isRequired="true"	はい	&columnname[1,1]
isRequired="false"	いいえ	&columnname[0,1]

次の例は、Siebel ビジネス・オブジェクト定義の一部分の図に示されている XSD ファイルに基づいて、DB2 コントロール・センターが作成する DDL を示しています。この図の XSD ファイルでは、isRequired 属性の値が false に指定されて組み込まれています。

```
CREATE NICKNAME sieb_ssa_Contact_Contact_NN(
  Id VARCHAR(15)  OPTIONS(XPATH './ns1:Id/text()',
    TEMPLATE '<ns1:Id>&column</ns1:Id>'),
  FirstName VARCHAR(50)  OPTIONS(XPATH './ns1:FirstName/text()',
    TEMPLATE '<ns1:FirstName>&column</ns1:FirstName>'),
  LastName VARCHAR(50)  OPTIONS(XPATH './ns1:LastName/text()',
    TEMPLATE '<ns1:LastName>&column</ns1:LastName>'),
  AccountId VARCHAR(255)  OPTIONS(XPATH './ns1:AccountId/text()'),
  PrimaryAccountName VARCHAR(100)
    OPTIONS(XPATH './ns1:PrimaryAccountName/text()'),
  PrimaryPostalCode VARCHAR(30)
    OPTIONS(XPATH './ns1:PrimaryPostalCode/text()'),
  PrimaryStreetAddress VARCHAR(200)
    OPTIONS(XPATH './ns1:PrimaryStreetAddress/text()'),
  SalesRep VARCHAR(255)  OPTIONS(XPATH './ns1:SalesRep/text()'),
  State VARCHAR(255)  OPTIONS(XPATH './ns1:State/text()')
  FOR SERVER siebel_server
  OPTIONS(XPATH '//ns1:sieb_ssa_Contact_Contact',
    TEMPLATE '<ns1:sieb_ssa_Contact_Contact>
      &Id[1,1] &FirstName[0,1] &LastName[0,1]
    </ns1:sieb_ssa_Contact_Contact>',
    BUSOBJ_NAME 'sieb_ssa_Contact_Contact',
    NAMESPACE 'ns1="http://www.ibm.com/websphere/
      crossworlds/2002/BOSchema/sieb_ssa_Contact_Contact"');
```

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』
- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 460 ページの『Web サービス・ラッパーのメッセージ』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ニックネームと XPATH 式

このトピックは、WebSphere® Business Integration ラッパーおよび Web サービス・ラッパーに適用されます。

ニックネームは、XML 文書データのツリー構造に対応しています。親ニックネームと子ニックネームは、データ・ツリー構造のルート構造とネスト・エレメントに対応します。それら親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。

各ニックネームは、出力値を表す XPath 式によって定義されます。WebSphere Business Integration ラッパーおよび Web サービス・ラッパーは、XPath 式を使用することによって、XML 文書のデータと、リレーショナル表の行との間の対応を確立します。それらの XPath 式は、XML 文書内の値を特定し、それらの値が各行の列にどう対応するかを決定します。WebSphere Business Integration ラッパーおよび Web サービス・ラッパーは、XML 文書データのみを読み取ります。ラッパーは、データを更新しません。XPath オプションには、SOAP エンベロープおよび SOAP ボディ・タグを通して SOAP メッセージを検索するための情報が含まれています。getQuote メッセージは、SOAP エンベロープおよびボディ・エレメントに含まれています。

NICKNAME オプション XPath 式は、出力エレメント中の繰り返しタグを指します。XPath 式は、ニックネーム中にいくつの行、またはどの行が含まれるかを決定します。列オプション XPath 式は、NICKNAME XPath 式に関係します。列オプション XPath は行中の値を示します。子ニックネーム中の NICKNAME オプション XPath は、親ニックネーム中の NICKNAME オプション XPath 式に関係します。

ニックネームを作成する際、そのニックネームと XML 文書との間の関連付けを指定するためのオプションを選択します。WebSphere Business Integration ラッパー用に作成されたニックネームは、XML スキーマ定義 (XSD) 文書と関連しています。Web サービス・ラッパー用に作成されたニックネームは、Web サービス記述言語 (WSDL) 文書と関連しています。

関連概念:

- 463 ページの『XML とは?』
- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』
- 471 ページの『ニックネームと XML 文書間のデータ関連』

関連タスク:

- 466 ページの『フェデレーテッド・システムへの XML の追加』
- 474 ページの『XML データ・ソースのニックネームの登録』

- 479 ページの『非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)』
- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成

ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューの作成は、ビジネス・アプリケーションをフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

ビジネス・オブジェクト階層を説明するニックネームの階層に対するフェデレーテッド・ビューを定義できます。フェデレーテッド・ビューを定義することにより、ビジネス・アプリケーション・ニックネーム階層の各部を結合する照会は正しく実行するようになります。

手順:

ビジネス・アプリケーション・ニックネームのフェデレーテッド・ビューを作成するには、次のようにします。

1. 各ビジネス・アプリケーション・ニックネームのビューを、親ニックネームへのパス上のすべてのニックネームの結合として定義する。
2. ビューの WHERE 文節に、結合述部として PRIMARY_KEY と FOREIGN_KEY 列を定義する。
3. ビューの SELECT リストに、ビジネス・アプリケーション・ニックネームのすべての列を組み込む。ただし、FOREIGN_KEY ニックネーム列オプションによって指定された列は除きます。階層とともに親ニックネームで PRIMARY_KEY として指定されている SELECT リスト内の列は組み込まないでください。
4. 階層の必須入力列を SELECT リストに組み込む。これらの列は、階層内で他のニックネームに属している可能性があります。

以下に、ビジネス・オブジェクトから生成されたニックネームに基づくビューの例を示します。 WHERE 文節には、CREATE NICKNAME ステートメントで定義されている主キーと外部キーが含まれています。

```
CREATE VIEW view1 (  
    customer, bankkey, bankact, customerno )  
AS (SELECT b.customer, b.bank_key, b.bank_acct,  
    a.customerno  
    FROM sap_bapi_customer_getdetail2_NN a,  
    sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b  
    WHERE a.NN__PKEY=b.NN__FKEY);
```

ビュー view1 を使用する照会には、次の例に示すように、必須列の述部値が組み込まれている必要があります。

```
SELECT * FROM view1  
    WHERE customerno='1234567890';
```

一連の作業はこれで終わりです。

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』

CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例

例 1: フラット・ビジネス・オブジェクト

157 ページの図 12 は、Siebel Business Component 用の WebSphere Business Integration ビジネス・オブジェクト定義を表す xsd ファイルの一部です。ビジネス・オブジェクト定義階層は単一レベルからなり、ルート・ビジネス・オブジェクトだけが含まれています。DB2 コントロール・センターは、このビジネス・オブジェクト定義を表す単一のリレーショナル・ニックネームを作成します。

xsd ファイル内では、注釈セクションに isRequired="true" フラグが追加されることにより、ID エlement には必須入力列のフラグが立てられています。FirstName および LastName 列は、isRequired="false" フラグが追加されることにより、オプション入力列として設定されています。

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://www.ibm.com/websphere/
  crossworlds/2002/BOSchema/sieb_ssa_Contact_Contact"
  ...
<xsd:element name="sieb_ssa_Contact_Contact">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version="1.0.0">
        <bx:appSpecificInfo>ON=Contact;CN=Contact</bx:appSpecificInfo>
      </bx:boDefinition>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:complexType>
<xsd:sequence>
  <xsd:element name="Id" minOccurs="0">
<xsd:annotation>
  <xsd:appinfo>
    <bx:boAttribute>
      <bx:appSpecificInfo>FN=Id</bx:appSpecificInfo>
      <bx:attributeInfo isForeignKey="false"
        isKey="true" isRequired="true" />
    </bx:boAttribute>
  </xsd:appinfo>
</xsd:annotation>
  ...
</xsd:simpleType>
</xsd:element>
<xsd:element name="FirstName" minOccurs="1">
<xsd:annotation>
  ...
  <bx:appSpecificInfo>FN=First Name</bx:appSpecificInfo>
  <bx:attributeInfo isForeignKey="false"
    isKey="false" isRequired="false" />
  ...
</xsd:annotation>
  ...
</xsd:element>
<xsd:element name="LastName" minOccurs="1">
<xsd:annotation>
  ...
  <bx:appSpecificInfo>FN=Last Name</bx:appSpecificInfo>
  <bx:attributeInfo isForeignKey="false"
    isKey="false" isRequired="false" />
  ...
</xsd:annotation>
  ...
</xsd:element>

```

図 12. フラット・ビジネス・オブジェクト用の XML スキーマ・ファイル (1/2)

1

```

<xsd:element name="AccountId" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:appSpecificInfo>FN=Account Id</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false"
      isKey="false" />
    ...
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="PrimaryAccountName" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:appSpecificInfo>FN=Primary Account Name</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false" isKey="false" />
    ...
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="PrimaryPostalCode" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:appSpecificInfo>FN=Primary Postal Code</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false" isKey="false" />
    ...
  </xsd:annotation>
</xsd:element>
<xsd:element name="PrimaryStreetAddress" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:appSpecificInfo>FN=Primary Street Address</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false" isKey="false" />
    ...
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="SalesRep" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:boAttribute>
    <bx:appSpecificInfo>FN=Sales Rep</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false" isKey="false" />
    ...
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="State" minOccurs="0">
  <xsd:annotation>
    ...
    <bx:boAttribute>
    <bx:appSpecificInfo>FN=State</bx:appSpecificInfo>
    <bx:attributeInfo isForeignKey="false" isKey="false" />
    ...
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="ObjectEventId"
  type="xsd:string" minOccurs="0" />
</xsd:sequence>
...
</xsd:schema>

```

図 12. フラット・ビジネス・オブジェクト用の XML スキーマ・ファイル (2/2)

DB2 コントロール・センターは、157 ページの図 12 に示されている XSD ファイルから、次の CREATE NICKNAME ステートメントを生成します。各入力列に TEMPLATE オプション値が指定されます。列オプション・テンプレートはニックネーム・オプション・テンプレートに関連付けられます。入力列も、ニックネーム・レベル TEMPLATE オプション値で参照されます。ニックネーム・オプション・テンプレートは入力ビジネス・オブジェクトの構造を指定します。ニックネーム・テンプレート値の各入力列参照の minOccurs 属性の値は、入力列が必須列かオプション列かを判別します。ID 列の参照は &Id[1,1] と指定されます。FirstName 列と LastName 列の参照は &FirstName [0,1] および &LastName [0,1] と指定されます。すべての出力列には XPATH 列オプション値が組み込まれています。ニックネームは、子をまったく含まないフラット・ビジネス・オブジェクト用です (カーディナリティー 'n' のエレメント)。

```
CREATE NICKNAME sieb_ssa_Contact_Contact_NN(
  Id VARCHAR(15)  OPTIONS(XPATH './ns1:Id/text()'),
    TEMPLATE '<ns1:Id>&column</ns1:Id>',
  FirstName VARCHAR(50)  OPTIONS(XPATH './ns1:FirstName/text()'),
    TEMPLATE '<ns1:FirstName>&column</ns1:FirstName>',
  LastName VARCHAR(50)  OPTIONS(XPATH './ns1:LastName/text()'),
    TEMPLATE '<ns1:LastName>&column</ns1:LastName>',
  AccountId VARCHAR(255)
    OPTIONS(XPATH './ns1:AccountId/text()'),
  PrimaryAccountName VARCHAR(100)
    OPTIONS(XPATH './ns1:PrimaryAccountName/text()'),
  PrimaryPostalCode VARCHAR(30)
    OPTIONS(XPATH './ns1:PrimaryPostalCode/text()'),
  PrimaryStreetAddress VARCHAR(200)
    OPTIONS(XPATH './ns1:PrimaryStreetAddress/text()'),
  SalesRep VARCHAR(255)  OPTIONS(XPATH './ns1:SalesRep/text()'),
  State VARCHAR(255)  OPTIONS(XPATH './ns1:State/text()')
  FOR SERVER siebel_server
  OPTIONS(XPATH '//ns1:sieb_ssa_Contact_Contact',
    TEMPLATE '<ns1:sieb_ssa_Contact_Contact>
      &Id[1,1] &FirstName[0,1] &LastName[0,1]
    </ns1:sieb_ssa_Contact_Contact>',
  BUSOBJ_NAME 'sieb_ssa_Contact_Contact',
  NAMESPACES 'ns1="http://www.ibm.com/websphere/
    crossworlds/2002/BOSchema/sieb_ssa_Contact_Contact"');
```

例 2: 階層ビジネス・オブジェクト

この例では、ビジネス・オブジェクト定義は、ルート・ビジネス・オブジェクトと 2 つの子ビジネス・オブジェクト (つまり合計 3 つの xsd ファイル) からなる 2 つのレベルの階層で構成されます。2 つのニックネームのみが生成されて、ビジネス・オブジェクト定義階層を表します。sap_customeraddress 子ビジネス・オブジェクトはカーディナリティーが 1 であり、そのことはエレメント定義に maxOccurs 属性指定がないことによって示されています。sap_customeraddress のすべての列は、ルート・ニックネーム sap_bapi_customer_getdetail2_NN に含まれています。sap_customerbankdetail 子ビジネス・オブジェクトはカーディナリティーが n であり、そのことはエレメント定義の maxOccurs="unbounded" 指定によって示されています。このオブジェクトは、別個の子ニックネーム sap_bapi_customer_getdetail2_sap_customerbankdetail_NN にマップされます。子ニックネームは、特殊な主キー外部キー関係によってルート・ニックネームに関連付けられています。

```

...
<xsd:element name="sap_bapi_customer_getdetail2">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version="3.0.0" />
    </xsd:appinfo>
  </xsd:annotation>

...
<xsd:element name="COMPANYCODE" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICOMPANYCODE:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>

...
</xsd:element>
<xsd:element name="CUSTOMERNO" minOccurs="1">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICUSTOMERNO:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>

...
</xsd:element>

```

図 13. SAP 階層ビジネス・オブジェクト: *customer_getdetail2* (1/2)

|

```

<xsd:element name="sap_customeraddress" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:ECUSTOMERADDRESS</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
        <bx:childObjectInfo relationship="Containment" version="3.0.0" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
<xsd:element ref="sap_customeraddress:sap_customeraddress" />
</xsd:sequence>

</xsd:complexType>
</xsd:element>
<xsd:element name="sap_customerbankdetail" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>
          ICUSTOMERBANKDETAIL:ECUSTOMERBANKDETAIL
        </bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
        <bx:childObjectInfo relationship="Containment" version="3.0.0" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="sap_customerbankdetail:sap_customerbankdetail"
        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="size" type="xsd:positiveInteger"
      default="1" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="ObjectEventId" type="xsd:string"
  minOccurs="0" />
</xsd:sequence>
...
<xsd:annotation>
  <xsd:appinfo>
    <bx:boVerb>
      <bx:appSpecificInfo>
        bapi.client.Bapi_customer_getdetail2
      </bx:appSpecificInfo>
    </bx:boVerb>
  </xsd:appinfo>
</xsd:annotation>
</xsd:enumeration>
<xsd:enumeration value="Update" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
...

```

図 13. SAP 階層ビジネス・オブジェクト: *customer_getdetail2* (2/2)

```

<xsd:element name="sap_customeraddress">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version="3.0.0">
        <bx:appSpecificInfo>:ECUSTOMERADDRESS</bx:appSpecificInfo>
      </bx:boDefinition>
    </xsd:appinfo>
  </xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="CUSTOMER" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:ECUSTOMER</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="NAME" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:ENAME</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="CITY" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:ECITY</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>

```

図 14. SAP 階層ビジネス・オブジェクト: customer_address (1/2)

1

```

<xsd:element name="POSTL_CODE" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:EPOSTL_CODE</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="STREET" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:ESTREET</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name="REGION" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>:EREGION</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="false" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
</xsd:sequence>
...

```

図 14. SAP 階層ビジネス・オブジェクト: *customer_address* (2/2)

1

```
...
<xsd:element name= "sap_customerbankdetail ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version= "3.0.0 ">
        <bx:appSpecificInfo>
          ICUSTOMERBANKDETAIL:ECUSTOMERBANKDETAIL
        </bx:appSpecificInfo>
      </bx:boDefinition>
    </xsd:appinfo>
  </xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name= "CUSTOMER " minOccurs= "0 ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICUSTOMER:ECUSTOMER</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey= "false " isKey= "true " />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
```

図 15. SAP 階層ビジネス・オブジェクト: *bank_detail* (1/2)

|

```

<xsd:element name= "BANK_KEY " minOccurs= "0 ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>IBANK_KEY:EBANK_KEY</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey= "false " isKey= "false " />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name= "BANK_ACCT " minOccurs= "0 ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>IBANK_ACCT:EBANK_ACCT</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey= "false " isKey= "false " />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name= "CTRL_KEY " minOccurs= "0 ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICTRL_KEY:ECTRL_KEY</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey= "false " isKey= "false " />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
<xsd:element name= "BANK_REF " minOccurs= "0 ">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>IBANK_REF:EBANK_REF</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey= "false " isKey= "false " />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
  ...

```

図 15. SAP 階層ビジネス・オブジェクト: bank_detail (2/2)

DB2 コントロール・センターは、166 ページの図 16 および 167 ページの図 17 に示されているように、3 つの SAP XSD ファイルから 2 つのニックネームを生成します。列 customerno は、customerno エレメントの XSD ファイル仕様のために、sap_bapi_customer_getdetail2_NN ニックネームのニックネーム・レベル・テンプレートで必須入力列としてマークされています。Customerno は I 接頭部と minOccurs=1 属性値でフラグが立てられています。

```

CREATE NICKNAME sap_bapi_customer_getdetail2_NN(
  CUSTOMER VARCHAR(10)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:CUSTOMER/text()'),
  NAME VARCHAR(35)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:NAME/text()'),
  CITY VARCHAR(35)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:CITY/text()'),
  POSTL_CODE VARCHAR(10)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:POSTL_CODE/text()'),
  STREET VARCHAR(35)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:STREET/text()'),
  REGION VARCHAR(3)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:REGION/text()'),
  NN_PKEY VARCHAR(16) FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'),
  COMPANYCODE VARCHAR(4) OPTIONS(XPATH './ns3:COMPANYCODE/text()',
    TEMPLATE '<ns3:COMPANYCODE>&column</ns3:COMPANYCODE>'),
  CUSTOMERNO VARCHAR(10) OPTIONS(XPATH './ns3:CUSTOMERNO/text()',
    TEMPLATE '<ns3:CUSTOMERNO>&column</ns3:CUSTOMERNO>'),
  ObjectEventId VARCHAR(48) OPTIONS(XPATH './ns3:ObjectEventId/text()'))
FOR SERVER sap_server
OPTIONS(XPATH '//ns3:sap_bapi_customer_getdetail2',
  TEMPLATE '<ns3:sap_bapi_customer_getdetail2>
    &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]
    &COMPANYCODE[0,1]
    &CUSTOMERNO[1,1]
  </ns3:sap_bapi_customer_getdetail2>',
  BUSOBJ_NAME 'sap_bapi_customer_getdetail2',
  NAMESPACES 'ns1="http://www.ibm.com/websphere/
    crossworlds/2002/BOSchema/sap_customeraddress",
    ns2="http://www.ibm.com/websphere/
    crossworlds/2002/BOSchema/sap_customerbankdetail",
    ns3="http://www.ibm.com/websphere/
    crossworlds/2002/BOSchema/sap_bapi_customer_getdetail2"');

```

図 16. *getdetail2* ニックネーム

|

```

CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10)  OPTIONS(XPATH './ns2:CUSTOMER/text()'),
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15)  OPTIONS(XPATH './ns2:BANK_KEY/text()'),
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18)  OPTIONS(XPATH './ns2:BANK_ACCT/text()'),
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2)   OPTIONS(XPATH './ns2:CTRL_KEY/text()'),
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20)  OPTIONS(XPATH './ns2:BANK_REF/text()'),
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16)  FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
  TEMPLATE '<ns3:sap_customerbankdetail>
  <ns2:sap_customerbankdetail>
  &CUSTOMER[0,1]
  &BANK_KEY[0,1]
  &BANK_ACCT[0,1]
  &CTRL_KEY[0,1]
  &BANK_REF[0,1]
  </ns2:sap_customerbankdetail>
  </ns3:sap_customerbankdetail>',
  NAMESPACE 'ns1="http://www.ibm.com/websphere/
  crossworlds/2002/BOSchema/sap_customeraddress",
  ns2="http://www.ibm.com/websphere/
  crossworlds/2002/BOSchema/sap_customerbankdetail",
  ns3="http://www.ibm.com/websphere/
  crossworlds/2002/BOSchema/sap_bapi_customer_getdetail2"');

```

図 17. 顧客銀行詳細ニックネーム

例 3: 主キーおよび外部キー

PRIMARY_KEY および FOREIGN_KEY 列は、親ニックネームと子ニックネームの間の関係を定義するために使用されます。各親ニックネームは、主キー列オプションをもっていなければなりません。ある親ニックネームの子は、親ニックネームの主キー列を参照する外部キー列オプションによって定義します。1 つのニックネームに対して複数の子は可能ですが、親については 1 つのみ可能です。

WebSphere Business Integration ラッパーの主キー値と外部キー値は、単一の照会内でのみ有効であり固有です。主キーと外部キーは、2 次照会を含む行の検索には使用できません。表に複数の照会のデータを追加されている場合には、値が固有であることが保証されないため、それらの値が保持されないのです。

以下の CREATE NICKNAME ステートメントは、160 ページの図 13、162 ページの図 14、および 164 ページの図 15 に示されている XML スキーマ定義ファイルから派生したものです。外部キー nn_fkey は子ニックネーム sap_bapi_customer_getdetail2_sap_customerbankdetail_nn を親ニックネーム sap_bapi_customer_getdetail2_nn と一意的に関連付けます。親ニックネームも、ニックネーム・オプション・テンプレート構造内の子ニックネームへの参照を使用します。

```

CREATE NICKNAME sap_bapi_customer_getdetail2_NN(
  CUSTOMER VARCHAR(10)
    OPTIONS(XPATH './ns3:sap_customeraddress/

```

```

        ns1:sap_customeraddress/ns1:CUSTOMER/text()'),
NAME VARCHAR(35)
  OPTIONS(XPATH './ns3:sap_customeraddress/
ns1:sap_customeraddress/ns1:NAME/text()'),
...
NN_PKEY VARCHAR(16) FOR BIT DATA NOT NULL
  OPTIONS(PRIMARY_KEY 'YES'),
...
TEMPLATE '<ns3:sap_bapi_customer_getdetail2>
  &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]
  &COMPANYCODE[0,1]
  &CUSTOMERNO[1,1]
</ns3:sap_bapi_customer_getdetail2>',
...
CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10) OPTIONS(XPATH './ns2:CUSTOMER/text()'),
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15) OPTIONS(XPATH './ns2:BANK_KEY/text()'),
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18) OPTIONS(XPATH './ns2:BANK_ACCT/text()'),
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2) OPTIONS(XPATH './ns2:CTRL_KEY/text()'),
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20) OPTIONS(XPATH './ns2:BANK_REF/text()'),
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16) FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
TEMPLATE '<ns3:sap_customerbankdetail>
  <ns2:sap_customerbankdetail>
    &CUSTOMER[0,1]
    &BANK_KEY[0,1]
    &BANK_ACCT[0,1]
    &CTRL_KEY[0,1]
    &BANK_REF[0,1]
  </ns2:sap_customerbankdetail>
</ns3:sap_customerbankdetail>',
...

```

例 4: ネーム・スペースを使用した XPath 式接頭部の解決

NAMESPACES オプションはコンマ区切りの名前と値の対のリストです。このオプションは、XPath 式に使用される接頭部を、XML スキーマに定義されているネーム・スペース URI を使用して解決します。これらの XPath 式は、WebSphere Business Integration アダプターから戻されるビジネス・オブジェクト (XML 文書) 上で適用されます。以下の式には、ネーム・スペース接頭部とその接頭部の定義が組み込まれています。

```

CREATE NICKNAME sap_customer
(
  sap_customeraddress_CUSTOMER VARCHAR(10)
    OPTIONS(XPATH './ns5:sap_customeraddress/
ns2:sap_customeraddress/ns2:CUSTOMER/text()'),
  sap_customeraddress_NAME VARCHAR(35)
    OPTIONS(XPATH './ns5:sap_customeraddress/
ns2:sap_customeraddress/ns2:NAME/text()'),
...
  sap_bapi_customer_getdet1_PKEY VARCHAR(16) FOR BIT DATA NOT NULL

```

```

    OPTIONS(PRIMARY_KEY 'YES'),
COMPANYCODE VARCHAR(4)
    OPTIONS(XPATH './ns5:COMPANYCODE/text()',
    TEMPLATE '<ns5:COMPANYCODE>&column</ns5:COMPANYCODE>'),
CUSTOMERNO VARCHAR(10)
    OPTIONS(XPATH './ns5:CUSTOMERNO/text()',
    TEMPLATE '<ns5:CUSTOMERNO>&column</ns5:CUSTOMERNO>'),
ObjectEventId VARCHAR(48)
    OPTIONS(XPATH './ns5:ObjectEventId/text()')
)
FOR SERVER SAP_SOURCE
    OPTIONS (
    XPATH '//ns5:sap_bapi_customer_getdetail2',
    TEMPLATE
        '<ns5:sap_bapi_customer_getdetail2>
        &customerbankdetail_NN[0,1] &COMPANYCODE[0,1] &CUSTOMERNO[1,1]
        </ns5:sap_bapi_customer_getdetail2>',
    BUSOBJ_NAME 'sap_bapi_customer_getdetail2',
    NAMESPACES '
ns2="http://www.ibm.com/websphere/
crossworlds/2002/BOSchema/sap_customeraddress",
...
ns5="http://www.ibm.com/websphere/
crossworlds/2002/BOSchema/sap_bapi_customer_getdetail2",
ns6="http://www.ibm.com/websphere/
crossworlds/2002/BOSchema/sap_return"
);

```

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項

等価述部

入力列で有効な述部は等価述部のみです。出力列では、どの述部も有効です。

以下の例では、この列では述部がサポートされていないことを示すメッセージと共にエラーが戻されます。この例では、列 `zipcode` は入力列です。

```
SELECT return FROM gettemp WHERE zipcode<'95141'
```

以下の例では、入力列に等価述部を使用した有効な照会を示します。カスタマーのニックネームは、カスタマー ID を含むローカル DB2 UDB 表で結合されています。照会には出力のみの列である `Sales` 列に追加の述部が含まれています。

```

SELECT a.name, a.address
      FROM customers a, local_table b
      WHERE
        a.customer_id=b.custid AND
        a.Sales > 300000;

```

必須入力列の述部

参照するニックネーム階層の SQL 照会では、すべての必須入力列に等価述部値を提供しなければなりません。ラッパーは、この制限に違反するすべての照会に対して、SQLCODE 901 を戻します。

IN または OR 述部

WebSphere Business Integration ラッパーおよび Web サービス・ラッパーでは、入力列に対して IN または OR 述部は許可されていません。

以下の例は、無効な照会を示しています。カスタマーのニックネームには 1 つの必須列 customer_id が含まれています。

```

SELECT * FROM customers
      WHERE customer_id IN (12345, 67890, 11223);
SELECT * FROM customers
      WHERE customer_id IN (SELECT custid FROM local_table); )

```

ただし、WebSphere Business Integration ラッパーの場合、必須入力列に SPECIFICATION ONLY パラメーターでユニーク索引を定義した場合、IN リスト述部を必須入力列と共に使用できます。

```

CREATE UNIQUE INDEX myuindex ON customers(customer_id) SPECIFICATION ONLY;

```

オプション入力列の結合

以下の例では、オプション入力列の結合の制限を説明します。オプション入力列をローカル表またはニックネームから結合することはできません。WSDL がオプションとして入力ニックネーム列を生成し、その列を結合に使用する必要がある場合には、DDL を編集してその列を必須入力列に変更しなければなりません。

この例では、order という名前の Web サービス・ラッパー・ニックネームが、shipping_method をオプション入力列として作成されます。以下のステートメントは、述部でリテラルを使用するため、有効な照会です。

```

SELECT * FROM order
      WHERE part="hammer" AND shipping_method="FEDEX";

```

ただし、部品と配送方法を定義する orderparts という名前のローカル表を照会に含め、その表にオプションの shipping_method という列が含まれる場合、ステートメントは無効です。

```

SELECT * FROM
      order o, orderparts op
      WHERE
        o.part="hammer" AND
        o.shipping_method=op.shipping_method

```

WebSphere Business Integration ラッパーの場合、ニックネームのオプション入力列の述部が WebSphere Business Integration アダプターにプッシュ・ダウンされる可能性があります。DB2 UDB は、アプリケーション・データ・ソースから取り出さ

れた行にそれらの述部をローカルに適用することができます。所定の入力列の述部が常にアダプターへプッシュ・ダウンされるようにするには、入力列を必須入力列と宣言してください。それぞれのニックネーム階層への照会には、必須入力列の述部値が含まれていなければなりません。

有効な結果を得るためには、結合された入力列は、Web サービス・ラッパーの必須列である必要があります。

外部結合

親ニックネームからの主キーおよび子ニックネーム列からの外部キーを使用したニックネーム間の外部結合はサポートされていません。

XML 文書中の親エレメントに子エレメントが含まれておらず、親ニックネームと子ニックネーム間で内部結合を使用した場合、そのエレメントには行が戻されません。たとえば、指定の顧客の `bankdetail` 情報が SAP システムにない場合、その特定の顧客の場合、

`sap_bapi_customer_getdetail2_sap_customerbankdetail_NN nickname` に行は戻されません。

以下の `CREATE NICKNAME` ステートメントは、例の照会で使用される列を定義します。

```
CREATE NICKNAME sap_bapi_customer_getdetail2_NN(  
  ...  
  NAME VARCHAR(35)  
    OPTIONS(XPATH './ns3:sap_customeraddress/  
      ns1:sap_customeraddress/ns1:NAME/text()'),  
  ...  
  NN_PKEY VARCHAR(16) OPTIONS(PRIMARY_KEY 'YES'),  
  COMPANYCODE VARCHAR(4) OPTIONS(XPATH './ns3:COMPANYCODE/text()'),  
    TEMPLATE '<ns3:COMPANYCODE>&column</ns3:COMPANYCODE>'),  
  CUSTOMERNO VARCHAR(10) OPTIONS(XPATH './ns3:CUSTOMERNO/text()'),  
    TEMPLATE '<ns3:CUSTOMERNO>&column</ns3:CUSTOMERNO>'),  
  ...  
  FOR SERVER sap_server  
  OPTIONS(XPATH '//ns3:sap_bapi_customer_getdetail2',  
  TEMPLATE '<ns3:sap_bapi_customer_getdetail2>  
    &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]  
    &COMPANYCODE[0,1]  
    &CUSTOMERNO[1,1]  
  </ns3:sap_bapi_customer_getdetail2>',  
  ...  
)
```

図 18. `getdetail2` ニックネームからの抜粋

```

CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10) OPTIONS(XPATH './ns2:CUSTOMER/text()'),
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15) OPTIONS(XPATH './ns2:BANK_KEY/text()'),
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18) OPTIONS(XPATH './ns2:BANK_ACCT/text()'),
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2) OPTIONS(XPATH './ns2:CTRL_KEY/text()'),
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20) OPTIONS(XPATH './ns2:BANK_REF/text()'),
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16) OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
  TEMPLATE '<ns3:sap_customerbankdetail>
  <ns2:sap_customerbankdetail>
    &CUSTOMER[0,1]
    &BANK_KEY[0,1]
    &BANK_ACCT[0,1]
    &CTRL_KEY[0,1]
    &BANK_REF[0,1]
  </ns2:sap_customerbankdetail>
</ns3:sap_customerbankdetail>',
  ...

```

図 19. カスタマー銀行詳細ニックネームからの抜粋

以下の例では、2 つのニックネーム間に内部結合条件があるため、照会は行を戻しません。

```

SELECT a.name, b.bank_key
FROM sap_bapi_customer_getdetail2_NN a,
     sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
WHERE a.customerno='1234567890'
AND a.NN_PKEY=b.NN_FKEY;

```

WebSphere Business Integration ラッパーまたは Web サービス・ラッパーのニックネーム定義に必須入力列が含まれる場合、このニックネームと他のローカル DB2 UDB 表または他のニックネームとの間の左外部結合はサポートされません。

関連概念:

- 149 ページの『ニックネームおよび列レベルでの TEMPLATE オプション』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』

ビジネス・アプリケーション・データ・ソース - 照会の例

例 1: 親ニックネームと子ニックネームの結合

SELECT ステートメントに子ニックネーム列が含まれている場合、親ニックネームとの結合述部を指定する必要があります。ニックネーム階層内で親子関係を維持するために、この結合条件が必要です。参照される子ニックネームから階層の親ニックネームへの階層関連を組み込むことによって、階層内の各親子ニックネームの対に主キーおよび外部キー結合条件を指定します。

ニックネーム階層を維持するために必要な要素すべてが含まれていないので、次の照会は無効です。

```
SELECT * FROM <child_nickname>;
SELECT b.col1
      FROM <parent_nickname> a,<child_nickname> b
      WHERE a.required_column=<value>;
```

次に示すのは、ニックネーム階層が維持された有効な照会の例です。

```
SELECT b.col1, a.col2
      FROM <parent_nickname> a,<child_nickname> b
      WHERE a.primary_key_column=b.foreign_key_column
            AND a.required_column=<value>;
```

次の例では、必須の親および子入力列がすべて WHERE 文節内の述部に組み込まれています。この WHERE 文節には、親主キー列と子外部キー列とが等しいことを指定する結合述部が含まれています。

```
SELECT a.customer, a.name, b.bank_key, b.bank_acct
      FROM sap_bapi_customer_getdetail2_NN a,
            sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
      WHERE a.nn_pkey=b.nn_fkey
            AND a.customer = 'ABC'
```

例 2: ビュー

2 種類のビューを作成することができます。

- 1 つは、子ニックネーム内の列から派生したビューで、照会に親子結合条件を組み込まなくても、直接子ニックネームに対して照会を発行できます。

```
CREATE VIEW view1 (
  customer, bankkey, bankact, customerno )
AS (SELECT b.customer, b.bank_key, b.bank_acct,
      a.customerno
      FROM sap_bapi_customer_getdetail2_NN a,
            sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
      WHERE a.NN_PKEY=b.NN_FKEY);
```

このビュー定義には、ニックネーム階層の必須入力列 customerno が組み込まれています。ビュー view1 を使用する照会には、次の例に示すように、必須列の述部値が組み込まれている必要があります。

```
SELECT * FROM view1
      WHERE customerno='1234567890';
```

- 主および外部キー列を除く、親および子ニックネームのすべての列を組み込むグローバル・ビュー。

例 3: 必須入力列

照会には、すべての必須入力列の述部値が組み込まれている必要があります。必須入力列は、ニックネーム `TEMPLATE` オプション値に `TEMPLATE` 列オプション定義と参照値 `[1,1]` が指定されている列です。ラッパーは、述部に必須入力列が組み込まれていない照会すべてに対してエラーを戻します。

次の例は、無効な照会を示しています。customers ニックネームに必須入力列 `customer_id` が含まれています。

```
SELECT * FROM customers;
```

次の例は、有効な照会を示しています。

```
SELECT * FROM customers WHERE customer_id = 123;
```

次の例は、表 `local_table` の `custid` 列にカスタマー ID を含む DB2 UDB 内のローカル表を示しています。この例は、WebSphere Business Integration ニックネームとローカル表との間の内部結合です。

```
SELECT a.name, a.address
       FROM customers a, local_table b
       WHERE a.customer_id=b.custid;
```

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』

関連資料:

- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

第 9 章 DB2 ファミリー・データ・ソースへのアクセスの構成

この章では、DB2 ファミリーのデータベースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。これらのデータベースには以下のものが含まれます。

- DB2 UDB for Linux、UNIX、および Windows
- DB2 UDB for z/OS and OS/390
- DB2 UDB for iSeries
- DB2 Server for VM and VSE

この章では、DB2 ファミリー・データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。DB2 ファミリー・データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

フェデレーテッド・サーバーへの DB2 ファミリー・データ・ソースの追加

フェデレーテッド・サーバーへの DB2 ファミリー・データ・ソースの追加

DB2 ファミリー・データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

DB2 コントロール・センターまたは DB2 コマンド行を使用して、DB2 ファミリー・データ・ソースへアクセスするようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- フェデレーション用に構成される DB2 サーバー。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

制約事項:

DB2 for Linux、UNIX、Windows バージョン 8.1 に保管されているデータにアクセスする場合は、DB2 データ・ソース別名のニックネームを作成できません。

手順:

DB2 データ・ソースをフェデレーテッド・サーバーに追加するには、次のようにします。

1. ノードをカタログします。
2. リモート・データベースをカタログします。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. DB2 サーバーとの接続をテストします。
7. DB2 の表およびビューのニックネームを登録します。

関連概念:

- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 49 ページの『FEDERATED パラメーターのチェック』
- 57 ページの『フェデレーテッド・データベースの作成』
- 176 ページの『フェデレーテッド・ノード・ディレクトリーへのノード項目のカタログ』
- 177 ページの『フェデレーテッド・システム・データベース・ディレクトリーへのリモート・データベースのカタログ』
- 178 ページの『DB2 ラッパーの登録』
- 180 ページの『DB2 データ・ソースのサーバー定義の登録』
- 182 ページの『DB2 データ・ソースのユーザー・マッピングの作成』
- 184 ページの『DB2 データ・ソース・サーバーへの接続のテスト』
- 185 ページの『DB2 表およびビューのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

フェデレーテッド・ノード・ディレクトリーへのノード項目のカタログ

フェデレーテッド・ノード・ディレクトリーへのノード項目のカタログは、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

DB2 データ・ソースのロケーションを指すには、フェデレーテッド・サーバーのノード・ディレクトリーに 1 つの項目をカタログします。フェデレーテッド・サーバーはこの項目を使用して、DB2 データ・ソースへの接続に使用する、適切なアクセス方式を決めます。

手順:

フェデレーテッド・ノード・ディレクトリーにノード項目をカタログするには、次のようにします。

1. 使用する通信プロトコルを判別します。
2. ノード項目をカタログするための適切なコマンドを発行します。
 - 通信プロトコルが Transmission Control Protocol/Internet Protocol (TCP/IP) の場合、**CATALOG TCPIP NODE** コマンドを発行します。

例:

```
CATALOG TCPIP NODE DB2NODE REMOTE SYSTEM42 SERVER DB2TCP42
```

DB2NODE 値は、カタログするノードに割り当てる名前です。 *REMOTE SYSTEM42* は、データ・ソースが存在するシステムのホスト名です。

SERVER DB2TCP42 は、サービス名または、サーバー・データベース・マネージャー・インスタンスの 1 次ポート番号です。サービス名を使用する場合は、大文字小文字の区別があります。

- 通信プロトコルが SNA の場合、**CATALOG APPC NODE** コマンドを発行します。

例:

```
CATALOG APPC NODE DB2NODE REMOTE DB2CPIC SECURITY PROGRAM
```

DB2NODE 値は、カタログするノードに割り当てる名前です。 *REMOTE*

DB2CPIC は、リモート・パートナー・ノードの SNA パートナー論理装置

(LU) 名です。 *SECURITY PROGRAM* は、パートナー LU に送信される割り振り要求に、ユーザー名とパスワードの両方を含めることを指定します。

この一連の作業における次のタスクは、『フェデレーテッド・システム・データベース・ディレクトリーへのリモート・データベースのカタログ』です。

関連タスク:

- 177 ページの『フェデレーテッド・システム・データベース・ディレクトリーへのリモート・データベースのカタログ』

フェデレーテッド・システム・データベース・ディレクトリーへのリモート・データベースのカタログ

フェデレーテッド・システム・データベース・ディレクトリーへのリモート・データベースのカタログは、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

リモート・データベースをフェデレーテッド・サーバー・システム・データベース・ディレクトリーにカタログすることによって、フェデレーテッド・サーバーが接続する DB2 データ・ソース・データベースを指定します。

手順:

リモート・データベースをフェデレーテッド・サーバー・システム・データベース・ディレクトリーにカタログするには、次のようにします。

1. クライアント構成アシスタント (CCA) を使用します。

UNIX のフェデレーテッド・サーバーの場合は、代わりに **CATALOG DATABASE** コマンドを使用できます。例:

```
CATALOG DATABASE DB2DB390 AS CLIENTS390 AT NODE DB2NODE AUTHENTICATION DCS
```

値 *DB2DB390* は、フェデレーテッド・サーバー・システム・データベース・ディレクトリーにカタログするリモート・データベースの名前です。 *AS CLIENTS390* は、カタログするデータベースの別名です。別名を指定していない場合、データベース・マネージャーはデータベース名 (例: *DB2DB390*) を別名として使用します。 *AT NODE DB2NODE* は、ノード・ディレクトリーにノード項目をカタログした時に指定したノード名です。 *AUTHENTICATION SERVER* は、DB2 データ・ソース・ノードで認証が行われることを指定します。

2. リモート・データベースの名前が 9 文字以上の場合、**CATALOG DCS DATABASE** コマンドを実行して、DCS ディレクトリー項目を作成する必要があります。例:

```
CATALOG DCS DATABASE SALES400 AS SALES_DB2DB400
```

値 *SALES400* は、カタログするリモート・データベースの別名です。この名前は、リモート・ノードに関連付けられたフェデレーテッド・サーバー・システム・データベース・ディレクトリーの項目名と一致しなければなりません。これは、**CATALOG DATABASE** コマンドで入力したのと同じ名前です。 *AS SALES_DB2DB400* は、カタログしたいターゲット・ホスト・データベースの名前です。

この一連の作業における次のタスクは、『DB2 ラッパーの登録』です。

関連タスク:

- 176 ページの『フェデレーテッド・ノード・ディレクトリーへのノード項目のカタログ』
- 178 ページの『DB2 ラッパーの登録』

DB2 ラッパーの登録

DB2 ラッパーの登録は、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

DB2 ファミリー・データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、**CREATE WRAPPER** ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER DRDA
```

推奨事項: DRDA というデフォルトのラッパー名を使用してください。このデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパー名に関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバーで、db2_wrapper という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER db2_wrapper LIBRARY 'libdb2drda.a'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『DB2 ラッパーのライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『DB2 ラッパーのサーバー定義の登録』です。

関連タスク:

- 180 ページの『DB2 データ・ソースのサーバー定義の登録』

関連資料:

- 179 ページの『DB2 ラッパーのライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

DB2 ラッパーのライブラリー・ファイル

次の表では、DB2 ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、libdb2drda.a、libdb2drdaF.a、および libdb2drdaU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 34. DB2 ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2drda.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2drda.sl
Linux	/opt/IBM/db2/V8.1/lib	libdb2drda.so

表 34. DB2 ラッパーのライブラリーのロケーションおよびファイル名 (続き)

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
Solaris	/opt/IBM/db2/V8.1/lib	libdb2drda.so
Windows	%DB2PATH%\bin	db2drda.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 178 ページの『DB2 ラッパーの登録』

DB2 データ・ソースのサーバー定義の登録

DB2 データ・ソースのサーバー定義の登録は、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスしたいそれぞれの DB2 サーバーを定義する必要があります。サーバー定義を登録すると、フェデレーテッド・サーバーはその DB2 サーバーに接続し、データベースにパッケージをバインドします。許可およびパスワードの情報はフェデレーテッド・グローバル・カタログに保管されていないので、それらの情報をサーバー定義に含める必要があります。

手順:

サーバー定義を DB2 データ・ソースに登録するには、CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE DB2/ZOS VERSION 6 WRAPPER DRDA
AUTHORIZATION "name1" PASSWORD "passwd1"
OPTIONS (DBNAME 'db_name')
```

サーバーに割り当てる名前は固有でなければなりません。重複するサーバー名は使用できません。

指定する VERSION オプションは、アクセスしたい DB2 データベース・サーバーのバージョンです。

WRAPPER パラメーターの名前は、CREATE WRAPPER ステートメントで指定した名前ではなければなりません。

データベース名は CREATE SERVER ステートメントではオプションとして指定しますが、DB2 データ・ソースの場合は必要です。

CREATE SERVER ステートメントを発行すると、フェデレーテッド・サーバーは DB2 データ・ソース・サーバーとの接続をテストします。

サーバー定義の登録後には、ALTER SERVER ステートメントを実行してサーバー・オプションを追加したりドロップしたりできます。

この一連の作業における次のタスクは、『DB2 データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 175 ページの『フェデレーテッド・サーバーへの DB2 ファミリー・データ・ソースの追加』
- 182 ページの『DB2 データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 6 ページの『サポートされるデータ・ソース』
- 181 ページの『CREATE SERVER ステートメント - DB2 ラッパーの例』

CREATE SERVER ステートメント - DB2 ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して DB2 ファミリー・データ・ソースのラッパー用サーバーを登録する方法の例が提供されます。このトピックには、すべての必要パラメーターでサーバーを作成する方法を示す完全指定の例が含まれます。

完全指定の例:

以下の例は、CREATE SERVER ステートメントを使用して DB2 ラッパーのサーバー定義を作成する方法を示しています。

```
CREATE SERVER DB2SERVER TYPE DB2/ZOS VERSION 6 WRAPPER DRDA
AUTHORIZATION "spalten" PASSWORD "db2guru"
OPTIONS (DBNAME 'CLIENTS390')
```

DB2SERVER

DB2 データベース・サーバーに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE DB2/ZOS

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。

VERSION 6

これは、アクセスしたい DB2 データベース・サーバーのバージョンです。

WRAPPER DRDA

CREATE WRAPPER ステートメントで指定した名前。

AUTHORIZATION "spalten"

これは、データ・ソース側での許可 ID です。この ID は、データ・ソース側の BINDADD 権限を持つ必要があります。この値は大文字小文字の区別があります。

PASSWORD "db2guru"

これは、データ・ソース側の許可 ID に関連したパスワードです。この値は大文字小文字の区別があります。

DBNAME 'CLIENTS390'

これは、アクセスしたい DB2 データベースの別名です。この別名は、**CATALOG DATABASE** コマンドを使用してデータベースをカタログした時に定義したものです。この値は大文字小文字の区別があります。

DB2 データ・ソースにはこのデータベース名が必要です。

サーバー・オプションの例:

サーバー定義の登録時に、**CREATE SERVER** ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと DB2 データ・ソースに特有のサーバー・オプションがあります。

以下の例は **CPU_RATIO** オプションを使用するサーバー定義を示します。

```
CREATE SERVER DB2SERVER TYPE DB2/ZOS VERSION 6 WRAPPER DRDA
AUTHORIZATION "spalten" PASSWORD "db2guru"
OPTIONS (DBNAME 'CLIENTS390', CPU_RATIO '0.001')
```

CPU_RATIO オプションを「0.001」に設定すると、これはリモート・データ・ソースの **CPU** がフェデレーテッド・サーバーより 1000 倍多くの容量を利用可能であることを示します。

関連タスク:

- 180 ページの『DB2 データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 623 ページの『付録 C. SQL ステートメントで有効なサーバーのタイプ』

DB2 データ・ソースのユーザー・マッピングの作成

DB2 データ・ソースのユーザー・マッピングの作成は、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

DB2 サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。フェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ローカル・ユーザー ID を DB2 サーバーのユーザー ID とパスワードにマップするには、**CREATE USER MAPPING** ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR USERID SERVER DB2SERVER
OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

REMOTE_AUTHID は接続許可 ID であり、バインド許可 ID ではありません。

この一連の作業における次のタスクは、『DB2 データ・ソース・サーバーへの接続のテスト』です。

関連タスク:

- 184 ページの『DB2 データ・ソース・サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 183 ページの『CREATE USER MAPPING ステートメント - DB2 ラッパーの例』

CREATE USER MAPPING ステートメント - DB2 ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、ローカルのユーザー ID を DB2 サーバーのユーザー ID およびパスワードにマップする方法を示す例が提供されます。このトピックには、すべての必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下の例は、ローカル・ユーザー ID を DB2 サーバー・ユーザー ID にマップする方法を示します。

```
CREATE USER MAPPING FOR DB2USER SERVER DB2SERVER
  OPTIONS (REMOTE_AUTHID 'db2admin', REMOTE_PASSWORD 'day2night')
```

DB2USER

DB2 ファミリー・データ・ソース・サーバーで定義されるユーザー ID にマップしようとしているローカルのユーザー ID を指定します。

SERVER DB2SERVER

CREATE SERVER ステートメントで定義した DB2 ファミリー・データ・ソース・サーバーの名前を指定します。

REMOTE_AUTHID 'db2admin'

DB2USER をマップする、DB2 ファミリー・データ・ソース・サーバー側の、接続許可ユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

REMOTE_PASSWORD 'day2night'

'db2admin' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

次の例は、特殊レジスター `USER` を含む `CREATE USER MAPPING` ステートメントです。

```
CREATE USER MAPPING FOR USER SERVER DB2SERVER
  OPTIONS (REMOTE_AUTHID 'db2admin', REMOTE_PASSWORD 'day2night')
```

DB2 特殊レジスター `USER` を使用して、`CREATE USER MAPPING` ステートメントを実行した人の許可 ID を、`REMOTE_AUTHID` ユーザー・オプションに指定されたデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 182 ページの『DB2 データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

DB2 データ・ソース・サーバーへの接続のテスト

DB2 データ・ソース・サーバーへの接続のテストは、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

サーバー定義および定義したユーザー・マッピングを使用して、DB2 サーバーへの接続をテストできます。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、DB2 システム表に対して SQL `SELECT` ステートメントを発行します。

例:

- DB2 for z/OS and OS/390 の場合:

```
SET PASSTHRU server_name
SELECT count(*) FROM sysibm.systables
SET PASSTHRU RESET
```

- DB2 for iSeries の場合:

```
SET PASSTHRU remote_server_name
SELECT count(*) FROM qsys2.systables
SET PASSTHRU RESET
```

SQL `SELECT` ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL `SELECT` ステートメントがエラーを戻す場合には、以下のことをする必要があります。

- リモート・サーバーが開始されているか確認します。
- リモート・サーバー上のリスナーを調べ、入ってくる接続に備えた構成がされているか確認します。

- ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD のオプションが DB2 サーバーとの接続に有効な設定になっているか確認します。
- DB2 カタログの、ノードとデータベースの項目をチェックします。
- DB2 フェデレーテッド・変数の設定をチェックして、リモート DB2 サーバーにアクセスできることを確認します。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) DB2COMM 変数が含まれます。
- サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。
- ユーザー・マッピングを調べます。必要であれば、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。

この一連の作業における次のタスクは、『DB2 表およびビューのニックネームの登録』です。

関連タスク:

- 185 ページの『DB2 表およびビューのニックネームの登録』
- 65 ページの『データ・ソース環境変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

DB2 表およびビューのニックネームの登録

DB2 表およびビューのニックネームの登録は、DB2 ファミリー・データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、CREATE NICKNAME ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録したときに収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに置かれます。オプティマイザーは、データ・ソースのカタログ情報の一部、またはすべてを使用するので、ニックネームを登録する前に、データ・ソース側で (RUNSTATS と同等のデータ・ソース・コマンドを使用して) 統計を更新しておくことをお勧めします。

DB2 ファミリー・データ・ソースにあるビューまたは表のニックネームを登録するには、CREATE NICKNAME ステートメントを使用します。DB2 ファミリー・データ・ソースを照会する時には、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用してください。

制約事項:

DB2 別名にニックネームを作成することはできません。

手順:

ニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME DB2NICKNAME FOR DB2SERVER.remote_schema.remote_table
```

ニックネームの長さは 128 文字までです。

ニックネームを登録したい DB2 表またはビューごとに、このステップを繰り返します。

ニックネームを登録すると、フェデレーテッド・サーバーは接続を使用してデータ・ソース・カタログを照会します。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連資料:

- 「コマンド・リファレンス」の『RUNSTATS コマンド』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 186 ページの『CREATE NICKNAME ステートメント - DB2 ラッパーの例』

CREATE NICKNAME ステートメント - DB2 ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して、アクセスしたい DB2 の表またはビューのニックネームを登録する方法の例が提供されます。

CREATE NICKNAME ステートメントの例を以下に示します。

```
CREATE NICKNAME DB2SALES FOR DB2SERVER.SALESDATA.EUROPE
```

DB2SALES

DB2 の表またはビューを識別するために使用する固有なニックネームです。

注: ニックネームは、スキーマとニックネームを組み込んだ 2 つの部分からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを作成したユーザーの許可 ID になります。

DB2SERVER.SALESDATA.EUROPE

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *DB2SERVER* は、CREATE SERVER ステートメントで DB2 データベース・サーバーに割り当てた名前です。
- *SALESDATA* は、表またはビューが属するリモート・スキーマの名前です。この値は大文字小文字の区別があります。
- *EUROPE* は、アクセスしたいリモートの表またはビューの名前です。

関連タスク:

- 575 ページの『ニックネームの変更』

- 185 ページの『DB2 表およびビューのニックネームの登録』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

第 10 章 Documentum データ・ソースへのアクセスの構成

この章では、Documentum データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。

Documentum データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- Documentum の概要を説明します。
- 実行する必要のあるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- Documentum ラッパーに関連するエラー・メッセージをリストします。

Documentum とは?

Documentum とは、文書の内容や属性の管理 (チェックイン、チェックアウト、ワークフロー、およびバージョン管理など) を行う文書管理ソフトウェアです。

Documentum 製品は 3 層であり、リレーショナル・データベースの最上部に構築されたクライアント/サーバー・システムです。

Docbase は、文書の内容、属性、関連、バージョン、レンダリング、フォーマット、ワークフロー、およびセキュリティーを保管する Documentum リポジトリです。拡張 SQL ダイアレクトである Documentum 照会言語 (DQL) は、Documentum データを照会するために使用されます。Docbase は、Oracle インスタンスまたは DB2[®] データベースに文書内容ファイルを加えたものと同じです。メタデータは、基礎となるリレーショナル・データベース管理システム (RDBMS) に保管され、内容はデータベースにバイナリー・ラージ・オブジェクト (BLOB) として保管されるか、サーバー・システムのファイル・システム内に保管されるファイルとして保管されます。Documentum についての詳細は、Documentum の資料を参照してください。

Documentum のラッパーを使用すると、Documentum データ・ソースを DB2 フェデレーテッド・システムに追加できます。Documentum データ・ソースをフェデレーテッド・システムに追加することによって、SQL ステートメントを使用して、Documentum Docbase 内のオブジェクトおよび登録済み表をアクセスしたり、照会したりすることができます。その後このデータをネイティブのデータ・ソースの外に移動せずに、フェデレーテッド・システム内の他のデータ・ソースと統合できます。Documentum ラッパーは、Documentum サーバーとのインターフェースとして、クライアント・ライブラリーを使用します。Documentum ラッパーは、2 つのバージョンの Documentum サーバーへのアクセスを提供します。それらは EDMS 98 (バージョン 3 と呼ばれる) と 4i です。190 ページの図 20 は、Documentum ラッパーがどのように機能するかを示しています。

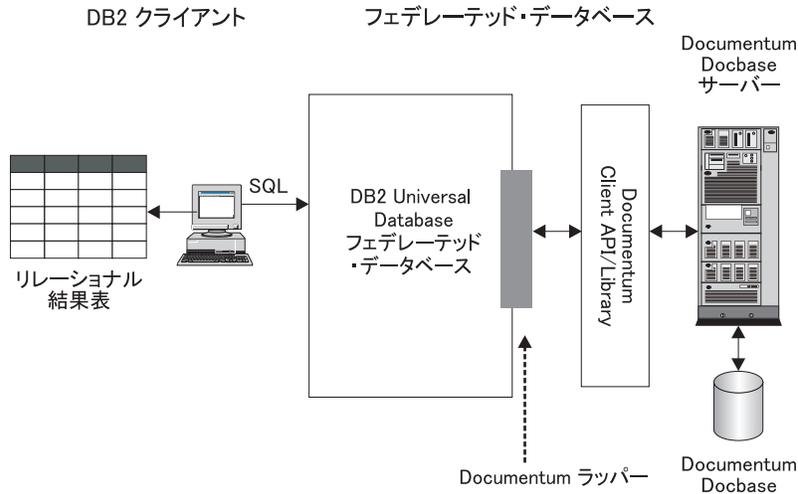


図 20. Documentum ラッパーがどのように機能するか

Documentum ラッパーが登録された後に、Documentum Docbase オブジェクトと登録済み表をリレーショナル表としてマップできます。これは、Docbase 属性を DB2 リレーショナル表の列名にマップすることによって行われます。

たとえば、表 35 は、Documentum Docbase のデフォルト文書タイプである dm_document の属性のサブセットを、関連するデータとともにリストしています。ここで、この属性サブセットが重要であると判断し、これらの属性をフェデレーテッド・データベース・システム内に接続するとします。このデータのサブセットの名前は DrugAB_data としました。

表 35. DrugAB_data

Title (タイトル)	Subject (件名)	Authors (作成者)	Keywords (キーワード)
ウサギに対する Drug A の効果	Drug A	Curran, L.	ウサギ, Drug A
Drug A の毒性結果	Drug A	Abelite, P., McMurtrey, K.	毒性, Drug A
Drug B の相互作用	Drug B	DeNiro, R., Stone, S.	相互作用, drug B
Drug B の化学構造	Drug B	Boyslim, F.	構造, drug B

Documentum ラッパーを登録し終わったら、SQL ステートメントを使用してデータを照会できます。

以下の示す照会では、subject (件名) が Drug A である title (タイトル) と authors (作成者) を表示します。結果の表を 表 36 に示します。

```
SELECT title, authors
FROM drugAB_data
WHERE subject = 'Drug A'
```

表 36. 照会の結果

Title (タイトル)	Authors (作成者)
ウサギに対する Drug A の効果	Curran, L.

表 36. 照会の結果 (続き)

Title (タイトル)	Authors (作成者)
Drug A の毒性結果	Abelite, P., McMurtrey, K.

関連タスク:

- 191 ページの『フェデレーテッド・サーバーへの Documentum データ・ソースの追加』

フェデレーテッド・サーバーへの Documentum の追加

フェデレーテッド・サーバーへの Documentum データ・ソースの追加

フェデレーテッド・サーバーが Documentum データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、フェデレーテッド・サーバーが Documentum データ・ソースにアクセスするように構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

Documentum データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. Documentum クライアント・ライブラリーをラッパーで使用可能にします。
2. Documentum 環境変数を設定します。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Documentum Docbase オブジェクトと登録済み表のニックネームを登録します。
7. Documentum ラッパーのカスタム関数を登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』

- 192 ページの『Documentum クライアント・ライブラリーのラッパーでの使用可能化』
- 193 ページの『Documentum 環境変数の設定』
- 194 ページの『Documentum ラッパーの登録』
- 195 ページの『Documentum データ・ソースのサーバーの登録』
- 196 ページの『Documentum データ・ソースのユーザー・マッピングの登録』
- 197 ページの『Documentum データ・ソースのニックネームの登録』
- 204 ページの『Documentum ラッパーのカスタム関数の登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Documentum クライアント・ライブラリーのラッパーでの使用可能化

Documentum クライアント・ライブラリーをラッパーで使用可能にすることは、Documentum をフェデレーテッド・システムに追加するという、より大きな作業の一部を成します。

ラッパーが正しく機能するには、ラッパーがクライアント・ライブラリーを使用できるようにしなければなりません。

前提条件:

Documentum ラッパーは、クライアント・ライブラリーのバージョン 3.1.7a を使用します。 Documentum 4i を使用する場合、Documentum からクライアント・ライブラリーの古いバージョンを入手する必要があります (まだインストールされていない場合)。

手順:

Documentum クライアント・ライブラリーをラッパーで使用可能にするには、シンボリック・リンクを作成するか、またはクライアント・ライブラリーをフェデレーテッド・サーバー上の適切なディレクトリーにコピーします。以下の表は、ライブラリーのコピー先とするディレクトリーをリストしています。

表 37. オペレーティング・システムごとのクライアント・ライブラリーおよびコピー先ディレクトリー

フェデレーテッド・サーバーのオペレーティング・システム	クライアント・ライブラリー	コピー先ディレクトリー
AIX	libdmcl.a	sql1lib/lib
Solaris	libdmcl.so	sql1lib/lib
Windows	dmcl32.dll	x:¥sql1lib¥bin

この一連の作業における次のタスクは、『Documentum ラッパー環境変数の設定』です。

関連タスク:

- 193 ページの『Documentum 環境変数の設定』

Documentum 環境変数の設定

Documentum 環境変数の設定は、Documentum をフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Documentum Docbases へのアクセスは、Documentum クライアント・ファイル `dmcl.ini` によって制御されます。フェデレーテッド・データベースのインスタンスは Documentum Docbase にアクセスできるようにするために、環境変数を Documentum クライアント・ファイル `dmcl.ini` に設定する必要があります。

Documentum 用の有効な環境変数は、以下のとおりです。

- DOCUMENTUM
- DMCL_CONFIG

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。『`db2dj.ini` ファイルの制約事項』

手順:

環境変数を設定するには、以下のように行います。

1. `db2dj.ini` ファイルを編集して、DOCUMENTUM または DMCL_CONFIG 環境変数のいずれかを設定します。
 - AIX および Solaris を実行するフェデレーテッド・サーバーでは、`db2dj.ini` ファイルは `$HOME/sql1lib/cfg` にあります。
 - Windows を実行するフェデレーテッド・サーバーでは、`db2dj.ini` ファイルは `x:%sql1lib%cfg` にあります (ここで **x:** は `sql1lib` ディレクトリーのあるドライブを表します)。

次の例は、UNIX オペレーティング・システム上でのこれらの変数の構文を表しています。

```
DOCUMENTUM=path
```

または

```
DMCL_CONFIG=path/dmcl.ini
```

ここで *path* は、使用したい `dmcl.ini` ファイルを含んでいる完全修飾ディレクトリーです。 `dmcl.ini` ファイルのパスについての詳細は、Documentum に付属の資料を参照してください。

| これらの両方の変数が db2dj.ini ファイル内で設定されている場合、
| DMCL_CONFIG 変数の値が使用されます。これらの変数のどちらも db2dj.ini
| ファイル内に設定しない場合、エラーが戻されます。

2. 図 21 に示されているように、DB2 インスタンスにアクセス可能なすべての Docbases の報告先となる docbroker の名前が dmcl.ini ファイルに指定されていることを確認してください。

```
##### DOCUMENTUM CLIENT CONFIGURATION FILE #####  
#  
# Copyright Documentum 1994.  
# Version 3.1 of the Documentum Server.  
#  
# A generated client init file for the Documentum Server.  
#  
# The only REQUIRED information in this file is the  
# [DOCBROKER_PRIMARY] section and an entry for host.  
# The host value should be the name of host on which  
# your network wide DocBroker is running  
  
[DOCBROKER_PRIMARY]  
host = server16.comp2.big.com
```

図 21. docbroker 名が指定されている dmcl.ini ファイルのサンプル

3. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。DB2 インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop  
db2start
```

この一連の作業における次のタスクは、『Documentum ラッパーの登録』です。

関連タスク:

- 194 ページの『Documentum ラッパーの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』

Documentum ラッパーの登録

Documentum ラッパーの登録は、Documentum データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Documentum データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上にラッパーを `dctm_wrapper` の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER dctm_wrapper LIBRARY 'libdb2lsdctm.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。`CREATE WRAPPER` ステートメントに指定する正しい名前については、『Documentum ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Documentum データ・ソースのサーバー定義の登録』です。

関連資料:

- 195 ページの『Documentum ラッパーのライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Documentum ラッパーのライブラリー・ファイル

次の表では、Documentum ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、`libdb2lsdctm.a`、`libdb2lsdctmF.a`、および `libdb2lsdctmU.a` です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 38. Documentum ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	<code>/usr/opt/db2_08_01/lib/</code>	<code>libdb2lsdctm.a</code>
Solaris	<code>/opt/IBM/db2/V8.1/lib</code>	<code>libdb2lsdctm.so</code>
Windows	<code>%DB2PATH%¥bin</code>	<code>db2lsdctm.dll</code>

`%DB2PATH%` は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは `C:¥Program Files¥IBM¥SQLLIB` です。

関連タスク:

- 194 ページの『Documentum ラッパーの登録』

Documentum データ・ソースのサーバーの登録

Documentum データ・ソースのサーバー定義の登録は、Documentum をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

制約事項:

同じ DB2 インスタンスで実行されているすべてのサーバーは、Documentum dmc1.ini ファイル内の同じ構成パラメーターを共有する必要があります。

手順:

サーバー定義は、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- このタスクを DB2 コントロール・センターから実行するには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、あるいは「サーバー定義 (Server Definitions)」フォルダーを右マウス・ボタンでクリックして、「作成」をクリックします。Discover ツールは、Documentum サーバーのノード名を検索します。サーバー定義を登録するには、RDBMS_TYPE および OS_TYPE サーバー・オプションの情報を指定する必要があります。
- このタスクを DB2 コマンド行から行うには、CREATE SERVER ステートメントを使用します。

例えば、AIX 上で実行する Docbase が含まれていてデータの保管に Oracle を使用する Documentum サーバーのために、サーバー定義 Dctm_Server1 を登録するには、次のステートメントを使用します。

```
CREATE SERVER Dctm_Server1
  TYPE DCTM
  VERSION 3
  WRAPPER Dctm_Wrapper
  OPTIONS( NODE 'Dctm_Docbase',
           OS_TYPE 'AIX',
           RDBMS_TYPE 'ORACLE');
```

この一連の作業における次のタスクは、『ユーザー・マッピングの作成』です。

関連タスク:

- 196 ページの『Documentum データ・ソースのユーザー・マッピングの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 592 ページの『CREATE SERVER ステートメントの引き数とオプション - Documentum ラッパー』

Documentum データ・ソースのユーザー・マッピングの登録

Documentum データ・ソースのユーザー・マッピングの作成は、Documentum をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

Documentum データ・ソースにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。フェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関

係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ユーザー・マッピングを作成するには、CREATE USER MAPPING ステートメントを使用します。

たとえば、次に示す CREATE USER MAPPING ステートメントは、Dctm_Server1 サーバーでユーザー Chuck をユーザー Charles にマップします。

```
CREATE USER MAPPING FOR Chuck SERVER Dctm_Server1
OPTIONS(REMOTE_AUTHID 'Charles', REMOTE_PASSWORD 'Charles_pw');
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行した人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されたデータ・ソースの許可 ID にマップすることができます。以下に例を示します。

```
CREATE USER MAPPING FOR USER SERVER Dctm_Server1
  OPTIONS(REMOTE_AUTHID 'Lisa', REMOTE_PASSWORD 'Lisa_pw');
```

この一連の作業における次のタスクは、『Documentum データ・ソースのニックネームの登録』です。

関連タスク:

- 197 ページの『Documentum データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 593 ページの『CREATE USER MAPPING ステートメントのオプション - Documentum ラッパー』

Documentum データ・ソースのニックネームの登録

Documentum データ・ソースのニックネームの登録

Documentum データ・ソースのニックネームの登録は、Documentum をフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

ユーザー・マッピングを作成した後に、ニックネームを登録する必要があります。定義するそれぞれの Documentum サーバーで、アクセスしたい各 Documentum Docbase オブジェクト・タイプまたは登録済み表ごとにニックネームを登録します。Documentum データ・ソースを照会するときには、Docbase の名前ではなく、これらのニックネームを使用します。

手順:

ニックネームを登録するには、CREATE NICKNAME ステートメントを使用します。

CREATE NICKNAME ステートメントの一部として、疑似列を定義できます。疑似列を定義しない場合、ラッパーがそれらを作成します。

この一連の作業における次のタスクは、『Documentum データ・ソースのカスタム関数の登録』です。

関連タスク:

- 204 ページの『Documentum ラッパーのカスタム関数の登録』
- 128 ページの『BLAST ラッパーで作業するための TurboBlast のセットアップ』

関連資料:

- 198 ページの『疑似列』
- 594 ページの『CREATE NICKNAME ステートメント構文 - Documentum ラッパー』
- 202 ページの『CREATE NICKNAME ステートメント - Documentum ラッパーの例』

疑似列

Documentum で使用する CREATE NICKNAME ステートメントでは、6 個の疑似列を定義します。これらの列は、オブジェクトの内容およびその他の情報にアクセスするために使用されます。

疑似列およびその定義は、表 39 にリストされています。

表 39. 疑似列名および定義

疑似列名	定義
GET_FILE	VARCHAR (nnn) ¹
GET_FILE_DEL	VARCHAR (nnn) ¹
GET_RENDITION	VARCHAR (nnn) ¹
GET_RENDITION_DEL	VARCHAR (nnn) ¹
HITS	INTEGER
SCORE	DOUBLE

注:

1. VARCHAR の長さは、フェデレーテッド・サーバーのオペレーティング・システムの最大パス長によって決まります。UNIX フェデレーテッド・サーバーの場合は、この長さは 1024 です。Windows フェデレーテッド・サーバーの場合は、この長さは 260 です。

表 40 は、SELECT 文節の疑似列をリストしたものです。

表 40. SELECT 文節の疑似列

疑似列名	説明
GET_FILE	<p>列の値のほかに、現在行の内容ファイルを検索します。</p> <p>内容ファイルの拡張子は、その Documentum の形式名です。同じ名前のファイルが存在する場合、それは上書きされます。</p> <p>GET_FILE は、オブジェクトの基本形式の取得を試みます。行の値は、ファイルの完全修飾ファイル名、または "no_content" という文字列です。</p> <p>たとえば、次のようになります。</p> <pre>SELECT object_name, get_file FROM ...</pre> <p>内容ファイルは、サーバーの CONTENT_DIR オプションで指定したサーバー・ディレクトリーに置かれます。また、ユーザーの DB2 ローカル名を使用して名前が付けられたサブディレクトリーにも置かれます。サブディレクトリーが存在しない場合は作成されます。</p> <p>その拡張子は、文書の形式タイプに対して Docbase で定義された DOS 拡張子になります。たとえば、MS Word 文書の場合は ".doc" です。</p>
GET_FILE_DEL	<p>GET_FILE_DEL がその照会において以前の行で検索したファイル (存在する場合) を最初に削除する点を除いて、この関数は GET_FILE と同種です。行の値は、ファイルの完全修飾ファイル名、または "no_content" という文字列です。</p>

表 40. SELECT 文節の疑似列 (続き)

疑似列名	説明
GET_RENDITION	<p>列の値のほかに、現在行で、その解釈 (オリジナル文書の異なる形式のコピー) の内容ファイルを検索します。</p> <p>内容ファイルの拡張子は、その Documentum の形式名です。同じ名前のファイルが存在する場合、それは上書きされます。</p> <p>解釈フォーマットを指定するためには、フォーム DCTM.RENDITION_FORMAT(<format> = 1 の述部を WHERE 文節に指定する必要があります。</p> <p>たとえば、次のようになります。</p> <pre>SELECT object_name, get_rendition FROM ... WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre> <p>GET_RENDITION は、オブジェクトの名前つき解釈の取得を試みます。行の値は、ファイルの完全修飾ファイル名、または "no_content" というストリングです。</p> <p>内容ファイルは、サーバーの CONTENT_DIR オプションで指定したサーバー・ディレクトリーに置かれます。また、ユーザーの DB2 ローカル名を使用して名前が付けられたサブディレクトリーにも置かれます。サブディレクトリーが存在しない場合は作成されます。</p> <p>その拡張子は、文書の形式タイプに対して Docbase で定義された DOS 拡張子になります。たとえば、MS Word 文書の場合は ".doc" です。</p>
GET_RENDITION_DEL	<p>GET_RENDITION_DEL がその照会において前の行で検索したファイル (存在する場合) を最初に削除する点を除いて、この関数は GET_RENDITION と同じです。行の値は、ファイルの完全修飾ファイル名、または "no_content" というストリングです。</p>

201 ページの表 41 は、検索文節を含む照会においての SELECT 文節の疑似列をリストしています。

表 41. 検索文節を含む、照会における *SELECT* 文節の疑似列

疑似列名	説明
HITS	<p>検索基準が一致した文書内の場所の数を表す整数を含みます。</p> <p>たとえば、次のようになります。</p> <pre>SELECT r_object_id, object_name, hits FROM std_doc WHERE DCTM.SEARCH_WORDS (''workflow'' OR ''flowchart'')=1</pre> <p>戻された各文書ごとに、文書の内容の中で "workflow" および "flowchart" というワードが出現する回数が合計され、HITS 値として戻されます。</p> <p>HITS 疑似列は、文書内に内容ファイルが 1 つしかない場合に適しています。これは、一般的な事例です。この疑似列は、<i>SELECT</i> ステートメントの <i>WHERE</i> 文節の修飾で使用できます。ただし、これは <i>SELECT</i> 文節でも指定する必要があります。</p>
SCORE	<p>文書の検索ヒット・ランキングを含みます。</p> <p>この疑似列は、Documentum の ACCRUE 概念演算子とともに使用します。両方とも、戻された各文書内で指定したワードが検出された数を示す数値を戻します。</p> <p>たとえば、次のようになります。</p> <pre>SELECT object_name, score FROM std_doc WHERE DCTM.SEARCH_TOPIC('<ACCRUE>("document","management","workflow")')=1 AND SCORE >=75</pre> <p>ステートメントは、内容の中に 2 つか 3 つの指定ワードを持つすべての文書を戻します。文書に 1 つのワードしかない場合、スコア 50 が割り当てられるため、<i>WHERE</i> 文節基準は失敗し、これは戻されません。3 つのうち 2 つのワードが検出される場合、文書にはスコア 75 が割り当てられます。3 つすべてのワードが検出される場合、文書のスコアは 88 になります。</p> <p>SCORE 疑似列は 1 つの内容ファイルを持つ文書に使用されます。これは、一般的な事例です。</p> <p>SCORE は、<i>WHERE</i> に <i>SEARCH_WORDS()</i> または <i>SEARCH_TOPIC()</i> 関数が含まれる場合のみ、<i>SELECT</i> 文節内に存在します。 <i>WHERE</i> 文節では、これは ACCRUE 概念演算子とともに使用されます。</p> <p>ACCRUE 概念演算子に関して詳しくは、Documentum の資料を参照してください。</p>

関連タスク:

- 197 ページの『Documentum データ・ソースのニックネームの登録』
- 204 ページの『Documentum ラッパーのカスタム関数の登録』

CREATE NICKNAME ステートメント - Documentum ラッパーの例

以下の CREATE NICKNAME ステートメントは、ニックネーム `std_doc` を定義します。Std_doc は、オブジェクト・タイプ `dm_document` を持つ Documentum Docbase と関連付けられています。表 42 は、Documentum 属性およびデータ・タイプを DB2 リレーショナル列名およびデータ・タイプにマップしています。これらは、CREATE NICKNAME ステートメントを構成するために使用されます。

表 42. std_doc ニックネームの DB2 列への Documentum 属性のマッピング

Documentum 属性名	Documentum データ・タイプ	DB2 列名	DB2 データ・タイプ	反復の有無	NULL 可能?
object_name	string(255)	object_name	varchar	いいえ	いいえ
r_object_id	ID	object_id	char(16)	いいえ	いいえ
r_object_type	string(32)	object_type	varchar	いいえ	いいえ
title	string(255)	title	varchar	いいえ	いいえ
subject	string(128)	subject	varchar	いいえ	いいえ
authors	string(32)	author	varchar	はい	はい
keywords	string(32)	keyword	varchar	はい	はい
r_creation_date	time	creation_date	timestamp	いいえ	はい
r_modify_date	time	modified_date	timestamp	いいえ	はい
a_status	string(16)	status	varchar	いいえ	いいえ
a_content_type	string(32)	content_type	varchar	いいえ	いいえ
r_content_size	double	content_size	integer	いいえ	いいえ
owner_name	string(32)	owner_name	varchar	いいえ	はい

表 43 は、ニックネームに使用される各 Documentum 属性を説明しています。

表 43. std_doc ニックネームの Documentum 属性の説明

Documentum 属性名	説明
object_name	オブジェクトのユーザー定義名。
r_object_id	このオブジェクトの固有オブジェクト ID。作成時に設定される。
r_object_type	オブジェクトのタイプ。オブジェクトの作成時に設定される。
title	オブジェクトのユーザー定義タイトル。
subject	オブジェクトのユーザー定義サブジェクト。
authors	オブジェクトに関する作成者のユーザー定義リスト
keywords	オブジェクトに関するユーザー定義キーワードのリスト
r_creation_date	オブジェクトが作成された日時。
r_modify_date	オブジェクトの最終変更日時。
a_status	ルーター・タスクが転送されるときにサーバーによって設定される。値は、ルーター・オブジェクト内の <code>attached_task_status</code> に対して割り当てられた値からとられる。
a_content_type	オブジェクトの内容のファイル形式。

表 43. *std_doc* ニックネームの *Documentum* 属性の説明 (続き)

Documentum 属性	
属性名	説明
<i>r_content_size</i>	内容のバイト数。複数ページの文書の場合、この属性は、文書と関連した最初の内容のサイズを記録する。
<i>owner_name</i>	オブジェクトの所有者の名前 (オブジェクトのを作成したユーザー)。

202 ページの表 42 は、以下の CREATE NICKNAME ステートメントに変換されます。

```
CREATE NICKNAME std_doc (
  object_name varchar(255) not null,
  object_id char(16) not null OPTIONS(REMOTE_NAME 'r_object_id'),
  object_type varchar(32) not null OPTIONS(REMOTE_NAME 'r_object_type'),
  title varchar(255) not null,
  subject varchar(128) not null,
  author varchar(32) OPTIONS(REMOTE_NAME 'authors', IS_REPEATING 'Y'),
  keyword varchar(32) OPTIONS(REMOTE_NAME 'keywords', IS_REPEATING 'Y'),
  creation_date timestamp OPTIONS(REMOTE_NAME 'r_creation_date'),
  modified_date timestamp OPTIONS(REMOTE_NAME 'r_modify_date'),
  status varchar(16) not null OPTIONS(REMOTE_NAME 'a_status'),
  content_type varchar(32) not null OPTIONS(REMOTE_NAME 'a_content_type'),
  content_size integer not null OPTIONS(REMOTE_NAME 'r_content_size'),
  owner_name varchar(32))
FOR SERVER Dctm_Server2 OPTIONS (REMOTE_OBJECT 'dm_document', IS_REG_TABLE 'N')
```

CREATE NICKNAME ステートメントをサブミットした後、ニックネーム *std_doc* を使用してフェデレーテッド・システムを照会することができます。フェデレーテッド・システム内で、*std_doc* ニックネームを他のニックネームと結合することもできます。

カタログでは、このニックネームの列数は、疑似列を含んでいるため CREATE NICKNAME ステートメントに指定されているものよりも 6 つだけ多くなります。

CreateNicknameFile ユーティリティを使用すると、Documentum タイプを自動的に DB2 タイプにマップしたり、初期 CREATE NICKNAME ステートメントを作成することができます。

関連タスク:

- 197 ページの『Documentum データ・ソースのニックネームの登録』

関連資料:

- 594 ページの『CREATE NICKNAME ステートメント構文 - Documentum ラッパー』

繰り返し属性の二重の定義 (Documentum ラッパー)

ラッパーの照会機能を最大限に活用するには、各属性を DB2 データ・タイプと同じように定義する必要があります。つまり、Documentum 整数を DB2 整数などとして定義する必要があります。しかし、このように定義すると、非 VARCHAR 繰り返し属性について複数の値が戻されなくなります。そのような列については、最後の値のみが戻されます。

この制限は、ラッパーが可能な場合いつでも、Docbase オブジェクトにつき結果として 1 つの行しか戻さないために生じます。この制限は、繰り返し属性が選択されている場合だけの問題です。しかし、同じリモートの繰り返し属性ではあっても、データ・タイプが VARCHAR である 2 次列を定義できます。

この列名は、すべての値の区切り文字で区切られたリストとして、すべての値を戻すために SELECT リストで使用されます。(各列の DELIMITER オプションは使用される区切り文字を指定します)

複数値の列のローカル名は標準化する必要があります。複数値の列のローカル名は、真のデータ・タイプとして定義される列のローカル名に、接頭部 "m_" を追加することによって標準化できます。

たとえば、データ・タイプ TIMESTAMP で定義されている、approval_dates という名前の Documentum 繰り返し属性のニックネーム列があるとします。この場合、m_approval_dates という名前の 2 次ニックネーム列を作成して、VARCHAR データ・タイプとして定義できます。さらに、SELECT リストで m_approval_dates を使用して、区切り文字で区切られたリストですべての承認日付を戻すことができます。

真のデータ・タイプが VARCHAR である繰り返し属性に関して、二重の定義を使用する必要はありません。

関連タスク:

- 197 ページの『Documentum データ・ソースのニックネームの登録』
- 575 ページの『ニックネームの変更』

Documentum データ・ソースの照会およびカスタム関数

Documentum ラッパーのカスタム関数の登録

Documentum データ・ソースのカスタム関数の登録は、Documentum をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。いくつかのカスタム関数を登録するには、CREATE FUNCTION ステートメントを使用する必要があります。これらの関数を使用して、照会内での全文検索や文書内容の検索などのいくつかの Documentum 固有な機能にアクセスすることができます。

述部のカスタム関数は 206 ページの表 44 にリストされています。

TOPIC 関数への参照は、Verity, Inc によってサード・パーティーの全文索引付けシステムの一部として提供される Documentum 関数への参照です。

制約事項:

DB2 はブール・タイプをサポートしていないため、WHERE 文節で使用されるほとんどのカスタム関数 (USER を除く) は "=1" についてチェックする必要があります。これらの関数は、整数を戻すために定義されているためです。

例:

```
"... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1"
```

手順:

カスタム関数を登録するには、CREATE FUNCTION ステートメントを使用します。

カスタム関数はすべて、DCTM というスキーマ名を使用して登録しなければなりません。各関数の完全修飾名は、DCTM.function_name です。

以下の例は、ANY_EQ カスタム関数を登録します。

```
CREATE FUNCTION DCTM.ANY_EQ (CHAR(), CHAR()) RETURNS INTEGER
AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION
```

各カスタム関数は、Documentum ラッパーをインストールしているそれぞれのフェデレーテッド・データベースごとに一度登録しなければなりません。

カスタム関数の登録するための助けとして、サンプル・ファイル create_function_mappings.ddl が sqllib/samples/lifesci/dctm ディレクトリーに提供されています。このファイルには、それぞれのカスタム関数ごとの定義が含まれています。この DDL ファイルを実行して、Documentum ラッパーをインストールしているそれぞれのフェデレーテッド・データベースごとにカスタム関数を登録することができます。

カスタム関数のストリング引き数の規則

ストリングとして渡される引き数はすべて、以下の規則に従います。

- 各ストリングは単一引用符で囲みます。
- ストリング内の単一引用符は、2 つの単一引用符によって表現されます。

照会でのカスタム関数の使用

以下の例は、照会でのカスタム関数の使用を示しています。

std_doc nickname から、1 つ以上の 'Dave Winters' という名前の作成者を持つ文書のオブジェクト名および作成者を表示するには、次のようにします。

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1
```

std_doc nickname から、1 つ以上の 'Dave Winters' または 'Jon Doe' という名前の作成者を持つ文書のオブジェクト名および作成者を表示するには、次のようにします。

```
SELECT object_name,authors FROM std_doc
WHERE DCTM.ANY_IN(authors,'Dave Winters','Jon Doe')=1
```

オブジェクト名および r_object_id を表示し、作成者の列に 'Dave Win%' というストリングを含む文書の std_doc ニックネームから内容ファイルを検索するには、次のようにします。

```
SELECT object_name, r_object_id, get_file FROM std_doc
WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1
```

カスタム関数表

表 44 は、述部のカスタム関数をリストしたものです。

表 44. 述部のカスタム関数

関数名	説明
ANY_EQ(arg1, arg2)	<p>指定した値と等しい任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_EQ(authors,'Dave Winters')=1</pre>
ANY_NE(arg1, arg2)	<p>指定した値と等しくない任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_NE(authors,'Dave Winters')=1</pre>
ANY_LT(arg1, arg2)	<p>指定した値より小さい任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_LT(num_approvers,4)=1</pre>
ANY_GT(arg1, arg2)	<p>指定した値より大きい任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_GT(num_approvers,3)=1</pre>
ANY_LE(arg1, arg2)	<p>指定した値以下の任意の値の反復属性をテストします。 2 つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_LE(num_approvers,2)=1</pre>

表 44. 述部のカスタム関数 (続き)

関数名	説明
ANY_GE(arg1, arg2)	<p>指定した値以上の任意の値の反復属性をテストします。2つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 比較される値を指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_GE(num_approvers,1)=1</pre>
ANY_IN(arg1, arg2 - arg11)	<p>指定した値リストの中の任意の10の値の反復属性をテストします。同じデータ・タイプの3～11の引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2-arg11 比較される値のコンマ区切りリストを指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_IN(authors,'Crick','Watson')=1</pre> <p>繰り返し属性の ANY_IN カスタム関数の値の最大値は、1つのステートメントにつき10です。複数のステートメントについては論理和演算されます。</p>
ANY_LIKE(arg1, arg2)	<p>指定した値と同種の任意の値の反復属性をテストします。2つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 単一引用符で囲まれたサブストリングと比較されるパターンを指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_LIKE(keywords,'%\$%')=1</pre> <p>ANY_LIKE() 述部ではエスケープ文節はサポートされません。</p>
ANY_NOT_LIKE(arg1, arg2)	<p>指定した値と異種の任意の値の反復属性をテストします。2つの必須引き数をとります。</p> <p>arg1 反復属性を表す列名を指定します。</p> <p>arg2 単一引用符で囲まれたサブストリングと比較されるパターンを指定します。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_NOT_LIKE(authors,'Dave Win%')=1 OR DCTM.ANY_NOT_LIKE(keywords,'%\$%')=1</pre> <p>ANY_NOT_LIKE() 述部ではエスケープ文節はサポートされません。</p>

表 44. 述部のカスタム関数 (続き)

関数名	説明
ANY_NULL(arg)	<p>IS NULL の反復属性をテストします。反復属性または単一値 DATE または TIMESTAMP 属性の名前となる 1 つの必須引き数をとります。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_NULL(authors)=1</pre>
ANY_NOT_NULL(arg)	<p>IS NOT NULL の反復属性をテストします。反復属性の名前となる 1 つの必須引き数をとります。</p> <p>例:</p> <pre>... WHERE DCTM.ANY_NOT_NULL(authors)=1</pre>
ANY_SAME_INDEX(arg1 - arg10)	<p>各属性の同じ索引で値の反復属性をテストします。2 ~ 10 の別の ANY_xx() 関数をとります。</p> <p>以下の例は、UCD に関連した少なくとも 1 人の Ken という作成者を文書が持つかどうかを調べます。</p> <pre>... WHERE DCTM.ANY_SAME_INDEX(ANY_EQ(author_name,'Ken'), DCTM.ANY_NE(author_affiliation,'UCD'))=1</pre> <p>繰り返し属性の同じ索引で行われる値のテストの最大数は、10 です。テストは、左から右へと検査される AND テストです。</p>
CABINET(arg) および CABINET_TREE(arg)	<p>Docbase キャビネットの完全修飾名となる 1 つの必須引き数をとります。</p> <p>例:</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 ... WHERE DCTM.CABINET_TREE('/MyDocs')=1</pre> <p>CABINET および CABINET_TREE の複数のインスタンスを使用して、複数のキャビネットを指定します。</p> <p>例:</p> <pre>... WHERE DCTM.CABINET('/Tools')=1 OR DCTM.CABINET_TREE('/Parts')=1</pre>
FOLDER(arg) および FOLDER_TREE(arg)	<p>Docbase フォルダーまたはキャビネットの完全修飾名となる 1 つの必須引き数をとります。</p> <p>例:</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 ... DCTM.FOLDER_TREE('/MyDocs/WhitePapers')=1</pre> <p>FOLDER および FOLDER_TREE の複数のインスタンスを使用して、複数のフォルダーを指定します。</p> <p>例:</p> <pre>... DCTM.FOLDER('/Tools/Drills')=1 OR DCTM.FOLDER_TREE('/Animals/Horses')=1</pre>

表 44. 述部のカスタム関数 (続き)

関数名	説明
RENDITION_FORMAT (フォーマット)	<p>GET_RENDITION および GET_RENDITION_DEL 疑似列を処理して、検索する解釈のフォーマットを設定します。フォーマットを指定する単一文字ストリング引き数をとります。</p> <p>次の例は、PDF フォーマットの文書を検索する例です。</p> <pre>SELECT get_rendition FROM WHERE DCTM.RENDITION_FORMAT('pdf')=1</pre>
USER(1)	<p>値を現行ユーザーの Documentum 作成者 ID と比較します。DB2 の制限のため、カスタム関数 USER は、使用されていない整数の引き数によって定義されます。</p> <p>例:</p> <pre>... WHERE approver = DCTM.USER(1)</pre> <p>DB2 作成者 ID に対応する Documentum 作成者 ID を作成するには、CREATE USER MAPPING ステートメントを使用します。</p>
SEARCH_WORDS(arg)	<p>個々のワードのリストとなる 1 つの必須ストリング引き数をとります。それらのワードは単一引用符で囲まれ、AND、OR、または NOT で区切られ、そして優先制御に対して括弧が使用されます。ワードに空白を含めることはできません。また、ワードは単一引用符で囲まなければなりません。</p> <p>例:</p> <pre>... DCTM.SEARCH_WORDS(''yeast'' AND (''bread'' OR ''cake'') AND NOT ''wedding'')=1</pre>
SEARCH_TOPIC(arg)	<p>Verity TOPIC 照会ステートメントとなる 1 つの必須ストリング引き数をとります。これは、Documentum および Verity に逐語的に渡されます。</p> <p>例:</p> <pre>... WHERE DCTM.SEARCH_TOPIC(''quick'')=1</pre>

一連の作業はこれで終わりです。

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE FUNCTION (ソースまたはテンプレート) ステートメント』

Documentum データ・ソース - 照会の例

ラッパーを登録したなら、Documentum データ・ソースに対して SQL 照会を実行できます。このセクションでは、いくつかの照会例が紹介されています。

照会を実行するには、通常の表名と表の列を使用する場合と同様に、SQL ステートメントでニックネームと定義済みのニックネーム列を使用します。

接続に関する制限事項:

DB2 アプリケーションによって確立される、DB2 データベースへの接続ごとに、Documentum ラッパーは最大で同時に 10 の Documentum セッションをサポートでき、各セッションでは最大で同時に 10 の Documentum 照会を管理できます。

1 つの DB2 アプリケーションで、同時に進行する複数の照会を実行できます。照会の存続期間は、DB2 にサブミットされるときに始まり、結果セット上の対応するカーソルが閉じるときに終了します。

指定された時間において、その時間に進行中の照会すべてに対して、1 つの Documentum サーバーから参照可能なニックネームは 10 以下です。複数の照会で参照される、または 1 つの照会で複数回参照されるニックネームは、参照回数ごとに 1 つとカウントされます。

LIKE 述部:

Documentum サーバーと DB2 とでは、LIKE 述部の処理方法が異なります。LIKE 述部が Documentum サーバーにプッシュダウンされると、Documentum セマンティクスが適用されるようになります。次の例で、列 c1 の内容が長さ 0 のストリングなら、Documentum の場合は述部が真 (true) に、DB2 の場合は偽 (false) になります。

```
c1 LIKE '%'
```

照会の例:

以下の照会は、'Test Document' という名前の文書の Docbase 文書すべてを表示します。

```
SELECT object_name
FROM std_doc
WHERE object_name='Test Document';
```

以下の照会は、カスタム関数 ANY_EQ を使用して、作成者の 1 人として 'Joe Doe' を持つすべての文書を表示します。

```
SELECT object_name
FROM std_doc
WHERE DCTM.ANY_EQ(author,'Joe Doe')=1
```

以下の照会は、FOLDER_TREE 関数および SEARCH_WORDS 関数を使用して、Approved キャビネット内のテキスト "protein" を含むすべての文書を検索します。

```
SELECT object_name
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.SEARCH_WORDS('protein')=1
```

以下の照会は、GET_FILE 疑似列、FOLDER_TREE カスタム関数、および ANY_IN カスタム関数を使用して、DB2 サーバー上のファイル名を検索します。この DB2 サーバーは、リストにあるすべての作成者を持つ、承認済みのキャビネット内のすべての文書の内容が配置されているサーバーです。

```
SELECT object_name, object_id, get_file
FROM std_doc
WHERE DCTM.FOLDER_TREE('/Approved')=1
      AND DCTM.ANY_IN(author, 'Mary Black', 'Joe Carson', 'Peter Miller')=1
```

関連タスク:

- 204 ページの『Documentum ラッパーのカスタム関数の登録』

関連資料:

- 203 ページの『繰り返し属性の二重の定義 (Documentum ラッパー)』
- 211 ページの『Documentum ラッパーのアクセス・コントロール』

Documentum ラッパーのアクセス・コントロール

照会は、Docbase における許可を得ていることが条件となります。最低でも読み取りアクセス権を付与されている文書のみが、照会の結果に含まれます。

関連資料:

- 404 ページの『表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル』
- 250 ページの『Excel ラッパーのファイル・アクセス・コントロール・モデル』

Documentum ラッパーのメッセージ

このセクションでは、Documentum のラッパーを処理する際に出されるメッセージをリストして説明します。

表 45. Documentum のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「sqlno_crule_save_plans [100]:rc (-2144272209) 空のプラン・リストが見つかった」。)	DB2 にサブミットされた SQL 照会はラッパーによって処理できませんでした。構文を訂正して再サブミットしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「dmAPI exec が失敗: [DM_QUERY_E_BAD_QUAL] エラー: 属性 <column_name> の属性修飾子、A0 は、正しい修飾子ではない」。)	REMOTE_OBJECT ニックネーム・オプションに不正な Documentum タイプまたは登録済みの表が入力されました。正しい Documentum タイプまたは登録済みの表を使用するようにニックネームを変更してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「無効な NULL 列が指定された」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。

表 45. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「ニックネーム指定が空」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「出力オブジェクトが空、または不完全」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「予期されない数の列が要求された」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「列情報が見つからない」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「サポートされない列タイプが要求された」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「正しくない列定義」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「不整合なタイプ; DB2 request != nickname type」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。

表 45. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「出力パラメーターが NULL ではない」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「照会出力変数が NULL ではない」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「無効なタイム・スタンプ長」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「列の数が矛盾する」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「値の変換時にデータにアクセスできなかった」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「DMCL クライアントの初期設定に失敗」。)	Documentum クライアントが初期設定できません。システム管理者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「Get_User が NULL を戻した」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。

表 45. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「Get_Local_User が NULL を戻した」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「開始トランザクションの失敗」。)	Documentum は begintrans が失敗したことを報告しました。システム管理者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「入力パラメーターが NULL ではない」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「Dctm 関数は DCTM.function(...) =1 のようではない」。)	Dctm 関数の述部の RHS として =1 が使用されませんでした。構文を変更して、照会をもう一度実行してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できます。 (理由「無効な列番号が要求された」。)	内部プログラミング・エラー。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL1881N	"DELIMITER" は "<column-name>" に対して有効な "COLUMN" オプションではありません。	DELIMITER オプションが列 <column-name> に指定されていましたが、IS_REPEATING オプションが指定されていませんでした。
SQL1882N	"SERVER" オプション "RDBMS_TYPE" は、"<server-name>" に対して "<option-value>" に設定できません。	RDBMS_TYPE サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません: DB2、 INFORMIX、 ORACLE、 SQLSERVER または SYBASE。
SQL1882N	"SERVER" オプション "TRANSACTIONS" は、"<server-name>" に対して "<option-value>" に設定できません。	TRANSACTIONS サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません: NONE、 QUERY、 PASSTHRU または ALL。

表 45. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"NICKNAME" オプション "IS_REG_TABLE" は、 "<nickname>" に対して "<option-value>" に設定できません。	IS_REG_TABLE ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません: 'Y' または 'N'。
SQL1882N	"NICKNAME" オプション "ALL_VERSIONS" は、 "<nickname>" に対して "<option-value>" に設定できません。	ALL_VERSIONS ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません: 'Y' または 'N'。
SQL1882N	"SERVER" オプション "OS_TYPE" は、"<server-name>" に対して "<option-value>" に設定できません。	OS_TYPE サーバー・オプションに指定された値が無効です。次の値のいずれかでなければなりません: AIX、HPUX、SOLARIS または WINDOWS。
SQL1882N	"NICKNAME" オプション "FOLDERS" は、"<nickname>" に対して "<option-value>" に設定できません。	FOLDERS ニックネーム・オプションに指定された値が無効です。これは、IS_REG_TABLE が 'Y' の表に指定することはできません。
SQL1882N	"NICKNAME" オプション "VERSIONS" は、"<nickname>" に対して "<option-value>" に設定できません。	VERSIONS ニックネーム・オプションに指定された値が無効です。次の値のいずれかでなければなりません: 'Y' または 'N'。 VERSIONS オプションは、IS_REG_TABLE オプションが 'Y' に設定されている表に対して 'Y' に設定できません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「無効な列名、IS_REG_TABLE、または IS_REPEATING が理由に指定された」	ニックネーム・ステートメントを調べ、IS_REG_TABLE、IS_REPEATING、REMOTE_NAME オプションおよび列名が正しく指定されているか確認してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「デバッグのためのログ・ファイルのオープンに失敗」	トラブルシューティング用に使用するログ・ファイルにアクセスできません。システム管理者に連絡してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「指定できるのは 1 つの検索条件だけ」	1 つの照会につき 1 つのカスタム検索機能しか指定できません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「内容ディレクトリーの作成に失敗」	DB2 エージェントによって宛先ディレクトリーが作成可能であることを確認してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「内容ファイルの許可の変更に失敗」	DB2 エージェントによって内容ディレクトリーが作成可能であることを確認してください。

表 45. Documentum のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL5182N	必須環境変数 "DMCL_CONFIG" が設定されていません。	DOCUMENTUM または DMCL_CONFIG 環境変数が設定されていませんでした。それらを db2dj.ini ファイルに設定してください。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 11 章 Entrez データ・ソースへのアクセスの構成

この章では、Entrez データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Entrez データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- Entrez の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- Entrez ラッパーに関連するエラー・メッセージをリストします。

Entrez とは?

Entrez は、National Center for Biotechnology Information (NCBI) の開発した照会および検索システムです。Entrez を使用すれば、NCBI がホストする複数のリンクされたデータベースにアクセスできます。

これらのデータベースには以下のものが含まれます。

- PubMed (生化学文献)
- Nucleotide (GenBank と呼ばれるシーケンス・データベース)
- OMIM (John Hopkins 大学による Online Mendelian Inheritance in Man)
- Genome (完全なゲノム組み立て)

Entrez のどのデータベースにも、Web ベースのある統一ツール・セットを使用してアクセスできます。Entrez ラッパーは、それらのツールを使用することによって、Entrez データベースを DB2[®] 環境に統合します。Entrez インターフェースでは数多くのデータベースがサポートされていますが、Entrez ラッパーでサポートされるのは PubMed と Nucleotide のみです。

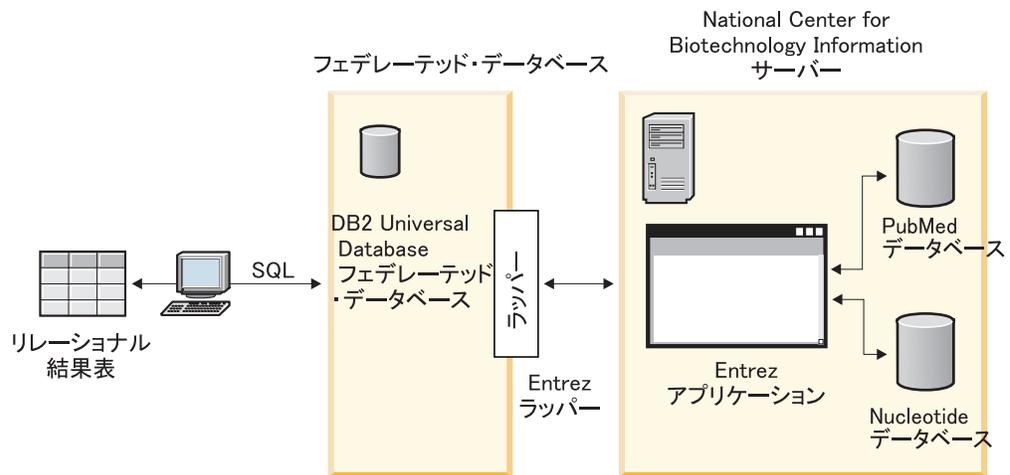


図 22. Entrez ラッパーがどのように機能するか

Entrez ラッパーを構成する要素の多くは、全データベースに共通です。そのような要素としては、次のものがあります。

- NCBI との Web 接続、および Entrez の ESearch および EFetch ユーティリティ
- 階層 XML データからリレーショナル表へのマッピング
- XML ラッパー・テクノロジーによる関連した表の結合

関連タスク:

- 218 ページの『フェデレーテッド・サーバーへの Entrez データ・ソースの追加』

フェデレーテッド・サーバーへの Entrez の追加

フェデレーテッド・サーバーへの Entrez データ・ソースの追加

フェデレーテッド・サーバーが Entrez データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、フェデレーテッド・サーバーが Entrez データ・ソースにアクセスすることができます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

Entrez ラッパーは、プロキシのあるファイアウォールを使用しているネットワーク中の PubMed および Nucleotide データ・ソースにアクセスできます。サポートされているプロキシは、HTTP、SOCKS4、および SOCKS5 です。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

Entrez のデータ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. Entrez ラッパーのカスタム関数を登録します。
2. ラッパーを登録します。
3. サーバー定義を登録します。
4. Entrez データベースのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 219 ページの『Entrez ラッパーのカスタム関数の登録』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Entrez ラッパーのカスタム関数の登録

Entrez ラッパーのカスタム関数の登録は、Entrez をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。カスタム関数を登録した後で、ラッパーを登録する必要があります。

制約事項:

- Entrez ラッパーのカスタム関数はすべて、`entrez` というスキーマ名を使用して登録する必要があります。
- 各カスタム関数は、Entrez ラッパーをインストールしているそれぞれの DB2 データベースごとに一度登録しなければなりません。

手順:

カスタム関数を登録するには、キーワード `AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION` を使用して `CREATE FUNCTION` ステートメントを発行してください。

各関数の完全修飾名は、`entrez.function_name` です。

以下の例では、`CONTAINS` 関数の 1 つのバージョンを登録します。

```
CREATE FUNCTION entrez.contains (varchar(), varchar())
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

カスタム関数を登録するには、サンプル・ファイル `create_function_mappings.ddl` を使用します。このサンプル・ファイルは、`samples/lifesci/entrez` ディレクトリ一の下、DB2 Information Integrator がインストールされているパスにあります。

サンプル・ファイルには、カスタム関数ごとの定義が含まれています。この DDL ファイルを実行して、Entrez ラッパーをインストールしているそれぞれのフェデレーテッド・データベースごとにカスタム関数を登録することができます。

この一連の作業における次のタスクは、『Entrez ラッパーの登録』です。

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE FUNCTION (ソースまたはテンプレート) ステートメント』
- 226 ページの『カスタム関数と Entrez 照会』
- 220 ページの『カスタム関数表 - Entrez ラッパー』

カスタム関数表 - Entrez ラッパー

Entrez カスタム関数を登録するには、CREATE FUNCTION ステートメントを使用します。

次の表では、Entrez カスタム関数およびその関数の登録時に指定する引き数のデータ・タイプがリストされています。関数で指定される最初の引き数は、タグ付きの列の列名に関するものです。関数に指定できる 2 番目の引き数は、検索語です。

表 46. Entrez ラッパーのカスタム関数

関数	説明
entrez.contains (varchar(), varchar())	指定する単語を使用してタグ付きの列を検索します。
entrez.contains (integer, varchar())	
entrez.contains (smallint, varchar())	
entrez.contains (real, varchar())	
entrez.contains (double, varchar())	
entrez.contains (date, varchar())	
entrez.contains (time, varchar())	
entrez.contains (char(), varchar())	
entrez.contains (timestamp(), varchar())	
entrez.search_term (char(), varchar())	

カスタム関数を登録するには、サンプル・ファイル create_function_mappings.ddl を使用します。サンプル・ファイルは samples/lifesci/entrez ディレクトリーにインストールされます。

関連タスク:

- 219 ページの『Entrez ラッパーのカスタム関数の登録』

関連資料:

- 226 ページの『カスタム関数と Entrez 照会』

Entrez ラッパーの登録

Entrez ラッパーの登録は、Entrez データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Entrez データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上にラッパーを `entrez_wrapper` の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER entrez_wrapper LIBRARY 'libdb2lsentrez.a'  
  OPTIONS(EMAIL 'jeff@someplace.com');
```

Entrez ラッパーを登録するには、E メール・アドレスを指定する必要があります。この E メール・アドレスはすべての照会に含められ、照会が多すぎて NCBI サーバーが過負荷になるなどの問題が生じる場合に、NCBI が連絡を取れるようにします。

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『Entrez ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Entrez データ・ソースのサーバー定義の登録』です。

関連資料:

- 221 ページの『Entrez ラッパーのライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Entrez ラッパーのライブラリー・ファイル

次の表では、Entrez ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、`libdb2lsentrez.a`、`libdb2lsentrezF.a`、および `libdb2lsentrezU.a` です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 47. Entrez ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパーのライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lsentrez.a
Linux	/opt/IBM/db2/V8.1/lib	libdb2lsentrez.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lsentrez.so
Windows	%DB2PATH%\%bin	db2lsentrez.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 220 ページの『Entrez ラッパーの登録』

Entrez データ・ソースのサーバーの登録

Entrez データ・ソースのサーバー定義の登録は、Entrez をフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

ラッパーの登録後、対応するサーバー定義を登録する必要があります。

特定のデータ・ソースによって表されているこのデータベース (PubMed または Nucleotide) は、CREATE SERVER ステートメントで示されているように、サーバー・タイプ値によって識別されます。このサーバー・タイプ値が、作成される任意のニックネームの構造を制御します。

手順:

Entrez サーバーをフェデレーテッド・システムに登録するには、CREATE SERVER ステートメントを発行します。

たとえば、entrez_wrapper ラッパーの pubmed_server1 という名前のサーバーを登録するには、次のステートメントを発行します。

```
CREATE SERVER pubmed_server1
  TYPE PUBMED
  VERSION 1.0
  WRAPPER entrez_wrapper;
```

さらに、entrez_wrapper ラッパーの nucleotid_server1 という名前のサーバーを登録するには、次のステートメントを発行します。

```
CREATE SERVER nucleotid_server1
  TYPE NUCLEOTIDE
  VERSION 1.0
  WRAPPER entrez_wrapper;
```

Entrez 照会で取り出す行数の制限

MAX_ROWS サーバー・オプションを使用して、Entrez ラッパーを使用した照会で戻される行数を制限できます。

SQL ステートメントの FETCH FIRST N ROWS ONLY 文節はユーザーまたはアプリケーションに戻される行数を制限しますが、MAX_ROWS サーバー・オプションを使用すると、この文節の場合とは違って、NCBI Web サイトから取り出す行数を制限できます。

MAX_ROWS オプション値は、照会で取り出せる行数の上限 (最大値) として常に使用されます。照会の際に MAX_ROWS オプションで指定された値より多くの行の取り出しが試行されると、結果セットは切り捨てられ、警告メッセージが出されま

す。

MAX_ROWS サーバー・オプションは、サーバーを作成する際に設定でき、ALTER SERVER ステートメントを使用して値を変更できます。

MAX_ROWS サーバー・オプションは必須ではありません。このオプションを設定しないと、デフォルト値が使用されます。使用される特定のデフォルト値は、ご使用のオペレーティング・システムによって異なります。Microsoft Windows オペレーティング・システムの場合は、デフォルト値は 2000 行です。UNIX ベースのオペレーティング・システムの場合は、デフォルト値は 5000 行です。

正数および 0 (ゼロ) のみ指定できます。このオプションを 0 (ゼロ) に設定すると、照会を使用して NCBI Web サイトから無制限の行数を取り出せます。しかし、MAX_ROWS サーバー・オプションを 0 (ゼロ) に設定したり、非常に大きな数値に設定したりすると、照会のパフォーマンスに影響することがあります。

プロキシ・サーバーを介して Entrez アクセスする

プロキシ・サーバーを介して Entrez データ・ソースにアクセスするには、サーバー定義を作成するときにオプションを指定する必要があります。指定するオプションは、アクセスしたいプロキシ・サーバーのタイプによって異なります。

HTTP プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、HTTP プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER pubmed_server_h
  TYPE PUBMED
  VERSION 1.0
  WRAPPER entrez_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_h',
          PROXY_SERVER_PORT '8080');
```

SOCKS4 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、SOCKS4 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER pubmed_server_s4
  TYPE PUBMED
  VERSION 1.0
  WRAPPER entrez_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS4', PROXY_SERVER_NAME 'proxy_4',
          PROXY_SERVER_PORT '1080');
```

認証情報のない SOCKS5 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、認証情報のない SOCKS5 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER pubmed_server_s5
  TYPE PUBMED
  VERSION 1.0
  WRAPPER entrez_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS5', PROXY_SERVER_NAME 'proxy_5',
          PROXY_SERVER_PORT '1081');
```

認証情報のある SOCKS5 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、認証情報のある SOCKS5 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER pubmed_server_s5a
  TYPE PUBMED
  VERSION 1.0
  WRAPPER entrez_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS5', PROXY_SERVER_NAME 'proxy_5',
          PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Khalid',
          PROXY_PASSWORD 'aaa', );
```

この一連の作業における次のタスクは、『Entrez データ・ソースのニックネームの登録』です。

関連タスク:

- 224 ページの『Entrez データ・ソースのニックネームの登録』

関連資料:

- 604 ページの『CREATE SERVER ステートメントの引き数 - Entrez ラッパー』

Entrez データ・ソースのニックネームの登録

Entrez データ・ソースのニックネームの登録は、Entrez をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

制約事項:

各 Entrez データベースのスキーマはラッパーによって固定されており、変更や修正はできません。それぞれのデータベースには、表ごとに固定された列のリストを伴う固定された表集合が存在します。データベース内の表には、階層関係があります。データベース内で他のすべての表の親である 1 つの表は、ルート (親) 表と呼ばれます。データベース内の他の表すべては、このルート表との間で親子関係を持っています。

手順:

Entrez データ・ソースのニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

ニックネーム列のリストはラッパーによって固定されて提供されるので、Nucleotide ニックネームを作成するための基本構文は単純です。例:

```

CREATE NICKNAME GBSeq FOR SERVER nucl;
CREATE NICKNAME GBFeatures FOR SERVER nucl;
CREATE NICKNAME GBIntervals FOR SERVER nucl;
CREATE NICKNAME GBQualifiers FOR SERVER nucl;
CREATE NICKNAME GBReference FOR SERVER nucl;

```

以下は、PubMed ニックネームを作成するための基本構文の例です。

```

CREATE NICKNAME pmarticles FOR SERVER pubmed_server;
CREATE NICKNAME PMACCESSION FOR SERVER pubmed_server;
CREATE NICKNAME PMCHEMICAL FOR SERVER pubmed_server;
CREATE NICKNAME PMMESH FOR SERVER pubmed_server;
CREATE NICKNAME PMCOMMENTS FOR SERVER pubmed_server;
CREATE NICKNAME PMARTICLEID FOR SERVER pubmed_server;

```

ニックネームの名前は、基本表の名前です。

この構文を使用すると、1 つの DB2 スキーマにつき 1 つのニックネーム・ファミリーのみを使用できます。REMOTE_OBJECT および PARENT ニックネーム・オプションを使用すると、他の名前を使用できます。ルート・ニックネームの場合には、REMOTE_OBJECT のみが必要となります。他のニックネームの場合、REMOTE_OBJECT および PARENT の両方が必要です。

次の例は、名前変更機能を使用した Nucleotide ニックネームの同一のセットを示しています。

```

CREATE NICKNAME NewSeq FOR SERVER nucl OPTIONS (REMOTE_OBJECT 'GBSEQ');
CREATE NICKNAME NewFeatures FOR SERVER nucl
  OPTIONS (REMOTE_OBJECT 'GBFEATURES', PARENT 'NEWSEQ');
CREATE NICKNAME NewIntervals FOR SERVER nucl
  OPTIONS (REMOTE_OBJECT 'GBINTERVALS', PARENT 'NEWFEATURES');
CREATE NICKNAME NewQualifiers FOR SERVER nucl
  OPTIONS (REMOTE_OBJECT 'GBQUALIFIERS', PARENT 'NEWFEATURES');
CREATE NICKNAME NewReference FOR SERVER nucl
  OPTIONS (REMOTE_OBJECT 'GBREFERENCE', PARENT 'NEWSEQ');

```

次の例では、名前変更機能を使用した PubMed ニックネームの同一のセットを示しています。

```

CREATE NICKNAME newpmarticles FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMARTICLES');
CREATE NICKNAME NEWPMACCESSION FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMACCESSION', PARENT 'NEWPMARTICLES');
CREATE NICKNAME NEWPMCHEMICAL FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMCHEMICAL', PARENT 'NEWPMARTICLES');
CREATE NICKNAME NEWPMMESH FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMMESH', PARENT 'NEWPMARTICLES');
CREATE NICKNAME NEWPMCOMMENTS FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMCOMMENTS', PARENT 'NEWPMARTICLES');
CREATE NICKNAME NEWPMARTICLEID FOR SERVER pubmed_server
  OPTIONS (REMOTE_OBJECT 'PMARTICLEID', PARENT 'NEWPMARTICLES');

```

この一連の作業における次のタスクは、『Entrez データ・ソースのカスタム関数の登録』です。

関連タスク:

- 218 ページの『フェデレーテッド・サーバーへの Entrez データ・ソースの追加』
- 219 ページの『Entrez ラッパーのカスタム関数の登録』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 230 ページの『PubMed スキーマ表』
- 235 ページの『Nucleotide スキーマ表』

Entrez データ・ソースの照会およびカスタム関数

カスタム関数と Entrez 照会

フェデレーテッド環境は 2 つの照会エンジンを使用します。Entrez ラッパーでは、これらの照会エンジンは DB2 および Entrez です。1 つの例外を除き、Entrez エンジン用のすべての述部はカスタム関数によって指定します。DB2 エンジンでは、関係演算子によってすべての述部を指定します。

主なカスタム関数は ENTREZ.CONTAINS です。CONTAINS 関数は検索語列引き数および照会テキスト引き数を必要とします。次の例は、ENTREZ.CONTAINS ステートメントを示しています。

```
ENTREZ.CONTAINS (search_term_column, query_text)
```

スキーマ・テーブルの Q 列にあるタグは、検索語を識別します。照会テキストは変更された Entrez 照会構文の中になければなりません。この構文は、ブール演算子 (OR、AND、および NOT) によって分離され、括弧によってグループ化された検索語で構成されています。CONTAINS 照会テキスト引き数の構文は標準 Entrez 照会構文とは異なり、検索語修飾子 (例: [pd]) は許可されていません。

カスタム関数は Entrez スキーマで登録されていて、関数を参照するにはこのスキーマを使用する必要があります。カスタム関数を使用する場合、その戻り値は等価述部にある値 1 と比較しなければなりません。

状態によっては、DB2 および Entrez の述部が処理できない仕方で混合されていることがあります。これらのケースでは、エラー・メッセージ SQL0142N ("SQL statement not supported") が生成されます。

たとえば、次の照会では、ラッパーが処理する述部の部分 (ENTREZ.CONTAINS 呼び出し) と DB2 が処理する必要のある部分 (BaseCountA の関係述部) とを分離することはできません。

```
WHERE  
  ENTREZ.CONTAINS (Organism, 'drosophila') = 1  
  OR (BaseCountA > 10 AND ENTREZ.CONTAINS (Keywords, 'glop') = 1)
```

いくつかの検索フィールドには、Entrez スキーマに対応する列がありません。たとえば、ヌクレオチド・データベースでは、[ALL] という語によってすべての検索可能フィールドが検索されますが、[WORD] という語によってレコードに関連するすべてのフリー・テキストが検索されます。これらの検索語のために疑似列が提供されています。疑似列が選択リストで参照されていると、NULL の値が戻されます。

ENTREZ.SEARCH_TERM マスター関数を実行すれば、別の方法では可能でないかもしれない照会を実行できます。ENTREZ.SEARCH_TERM マスター関数を指定する場合、その関数は照会の中にあるただ 1 つのカスタム関数である必要があります。それぞれの照会ごとに、Entrez ニックネーム 1 つにつきただ 1 つの

ENTREZ.SEARCH_TERM マスター関数の存在が可能です。さらに、SEARCH_TERM と CONTAINS 関数は同じ照会の中にある同じニックネームについて混合することはできません。最初の引き数である列仕様は、親のニックネームの主キー列である必要があります。2番目の引き数である照会テキストは、検索フィールド修飾子を組み込んだ Entrez 書式検索用語です。このテキストは、URI 構文が必要とされる URI エスケープを除き、変更されずに Entrez に受け渡されます。

次の例は、PubMed ニックネーム の WHERE 文節のある照会を示します。

```
WHERE
  ENTREZ.CONTAINS (authors, 'kaufmann OR ito AND NOT rakesh')
  AND
  (ENTREZ.CONTAINS (title, 'drosophila')
   OR
   ENTREZ.CONTAINS(alltext, 'drosophila OR "fruit fly"'))
```

この例では、個々の述部は authors、title、および all text です。

個々の述部が変更され、修飾子が各検索語の後に付け加えられます。その後、その語が括弧でグループ化され、DB2 ブール演算子優先順位が施行されます。これらの変更のため、authors 述部は次のようになります。

```
((kaufmann[auth] OR ito[auth]) AND (NOT (rakesh[auth])))
```

title 述部は次のようになります。

```
(drosophila[title])
```

そして、all text 述部は次のようになります。

```
(drosophila[all] OR "fruit fly"[all])
```

個々の述部が結合されるとき、DB2 ブール演算子優先順位を保守するために括弧が使用されます。ストリングを URI の一部として表現するのに必要なテキスト・トランスフォーメーションを除外して、Entrez にサブミットされる最終検索語ストリングは次のとおりです。

```
((kaufman[auth] OR ito[auth]) AND (NOT (rakesh[auth]))) AND
((drosophila[title]) OR (drosophila[all] OR "fruit fly"[all]))
```

関連資料:

- 220 ページの『カスタム関数表 - Entrez ラッパー』

Entrez ラッパーの関係述部

Entrez ラッパーでは、ニックネーム列において =、BETWEEN、LIKE、<> などの関係述部がサポートされています。しかし、Entrez の検索エンジンが処理するのは、それらの関係述部のうちの一部のみです。Entrez 検索エンジンで処理されない関係述部は、DB2[®] によって処理されます。Entrez 検索エンジンは、スキーマごとに特定の ID 列に関する等号 (=) 述部および IN 述部を処理します。それらの述部を使用すれば、Entrez ラッパーによる検索フェーズを実行することなく、直接フェッチ・フェーズを実行できます。有効な述部には、たとえば次のものがあります。

```
WHERE pmid = '1234567'
```

```
WHERE medlineid IN ('1234567', '9191919')
```

この種の述部に使用できる列は、スキーマ表の F 列によって識別されているものです。このオプションの値は Y でなければなりません。

関連概念:

- 228 ページの『Entrez ラッパーで無効な WHERE 文節』

関連タスク:

- 229 ページの『Entrez データ・ソース - 照会の例』
- 219 ページの『Entrez ラッパーのカスタム関数の登録』

Entrez ラッパーで無効な WHERE 文節

Entrez ラッパーは、結果として NCBI データベースの非修飾スキャンが実行されることになるような照会をリジェクトします。有効な WHERE 文節には、スキーマの主 ID に対する等号 (または IN) 述部か、またはカスタム関数が含まれていなければなりません。これらの基準を満たさない照会は、エラー・コード SQL0142N または SQL30090N でリジェクトされます。

関連概念:

- 227 ページの『Entrez ラッパーの関係述部』

関連タスク:

- 229 ページの『Entrez データ・ソース - 照会の例』
- 219 ページの『Entrez ラッパーのカスタム関数の登録』

スキーマ・データ・エレメントの単純化

いくつかのデータ・エレメントは、SQL スキーマによって表現される際に標準形式に変換されます。そのようなデータ・エレメントとしては、項目リスト、名前、および日付があります。

項目リスト

特に指定されていない限り、項目のリストを非標準化して単一の列にすると、その個々の項目はセミコロンと 1 個のスペースで区切られます。たとえば、項目に dnaA gene、dnaN gene、および orf187 の各キーワードが含まれている場合、対応する Keywords 列の内容は「dnaA gene; dnaN gene; orf187」になります。

名前

NCBI スキーマに含まれる名前は、姓 (必須) と、いくつかのオプション・エレメントのうちの 1 つで構成されます。それらのオプション・エレメントの中には一緒に出現するものと、互いに排他的なものがあります。名前の標準形式を作成するには、それらのエレメントに優先順位を割り当てます。優先順序の高いものから順にエレメントを並べると、次のようになります。

- Forename (名)
- First name (ファーストネーム) または middle name (ミドル・ネーム)
- Initials (イニシャル)

名前は、所属付きで表示したり所属なしで表示したりできます。所属なしの場合、名前の形式は「<last name>, <first>」となります。ここで <first> は、オプション・エレメントのいずれかです。 <first> エレメントが見つからない場合は、コンマは使用されません。所属は、(<affiliation>) の形式で追加できます。

非標準化リストの氏名は、セミコロンとスペースで区切ります。名前を区切る正しい方法の一例を次に示します。

Parker, M. J.; Ranjan, K. A.

日付

日付 (特に出版日付) は、NCBI スキーマのさまざまな形式で出現します。それらのさまざまな形式に対応しつつ、日付の比較や可能な範囲で日付の算術演算ができるようにするため、SQL スキーマの日付は 2 種類の形式で表されることになっています。まず、日付は文字ストリングとして指定できます。第 2 に、日付は DATE タイプの列として指定できます。

日付値に月しか含まれておらず、日のデータが参照されていない場合、デフォルトの日として月の最初の日が使用されます。月 (または月日) ではなく季節が指定されているなら、その季節の最初の日が使用されます。

関連タスク:

- 224 ページの『Entrez データ・ソースのニックネームの登録』

関連資料:

- 226 ページの『カスタム関数と Entrez 照会』

Entrez データ・ソース - 照会の例

このトピックでは、Entrez データ・ソースに対して実行する照会の例をいくつか取り上げます。

手順:

照会を実行するために、ガイドとして以下の例を使用します。

PubMed ニックネームにおいて:

以下に、PubMed ニックネームで、単一のフェッチ・キーを使用する照会を示します。

```
select PMID, ArticleTitle FROM pmarticles WHERE pmid = '12345';
```

以下に、PubMed ニックネームで、混合されたフェッチ・キーを使用する照会を示します。

```
select PMID, ArticleTitle FROM pmarticles
WHERE pmid = '12345' OR MedlineID = '12346';
```

以下に、PubMed ニックネームで、CONTAINS 関数を使用する照会を示します。

```
select PMID, ArticleTitle FROM pmarticles
WHERE entrez.contains (ArticleTitle, 'granulation') = 1
AND entrez.contains (PubDate, '1992') = 1;
```

以下に、PubMed ニックネームで、指定された AuthorList および LanguageList を検索する照会を示します。

```
select PMID, ArticleTitle FROM pmarticles
WHERE entrez.contains (AuthorList, 'Albarrak') = 1
AND entrez.contains (LanguageList, 'eng')=1;
```

以下に、PubMed ニックネームで、複合述部を使用する照会を示します。

```
select PMID, ArticleTitle FROM pmarticles
WHERE entrez.contains (PublicationTypeList, 'Journal Article') = 1
AND entrez.contains (MedlineTA, 'sun')=1
OR entrez.contains (PersonalNameSubjectList, 'shine')=1;
```

Nucleotide ニックネームにおいて:

以下に、Nucleotide ニックネームで、複数のフェッチ・キーを使用する照会を示します。

```
select PrimaryAccession, LocusName, SeqLength from gbseq
WHERE PrimaryAccession in ('NM_000890', 'NC_003106');
```

以下に、Nucleotide ニックネームで、検索可能なフィールドすべてを検索する照会を示します。

```
select PrimaryAccession, substr(Definition,1,300), GI from gbseq
WHERE entrez.contains(AllText, 'abcde')=1;
```

以下に、Nucleotide ニックネームで、すべてのフリー・テキストを検索する照会を示します。

```
select * from gbseq WHERE entrez.contains(FreeText, 'abcde')=1;
```

以下に、Nucleotide ニックネームで、ある定義を検索する照会を示します。

```
select PrimaryAccession, substr(Definition,1,300), version, GI from gbseq
WHERE entrez.contains(Definition, 'Sulfolobus tokodaii
AND complete genome') = 1;
```

以下に、Nucleotide ニックネームで、あるキーワードを検索する照会を示します。

```
select PrimaryAccession, substr(KeywordList,1,200), Segment from gbseq
WHERE entrez.contains(KeywordList, 'nkcc1 gene') = 1;
```

関連概念:

- 227 ページの『Entrez ラッパーの関係述部』
- 228 ページの『Entrez ラッパーで無効な WHERE 文節』

関連タスク:

- 219 ページの『Entrez ラッパーのカスタム関数の登録』

PubMed スキーマ表

| PubMed スキーマは、PubMed タイプのサーバーから得られるデータの表示方法を定義します。このスキーマは、以下の関連する複数のニックネームで構成されています。

- | • PMArticles
- | • PMAccession
- | • PMChemical

- PMMeSHHeading
- PMComments
- PMArticleID

以下の表では、各ニックネームに含まれる列についての情報をリストします。「タグ」列には、その列に対する有効な検索タグが含まれています。有効な探索タグのリストについては、以下の Web サイトを参照して、「Search Field Descriptions and Tags」へのリンクを見つけてください。

www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html

ニックネームを作成する場合、列に関するデフォルト・データ・タイプをオーバーライドできます。Entrez ラッパーは、最大で 5 メガバイトまでの長さの CLOB データ・タイプをサポートしています。

ニックネームを作成するときに、列のデフォルトの長さをオーバーライドできます。例えば、PMArticles ニックネーム内の Abstract 列などいくつかの列は、大量のデータを戻すことがあります。この列のデフォルトの長さは、VARCHAR(32000) です。列の最初の 100 バイトを戻すには、列にデータ・タイプ VARCHAR(100) を定義します。これにより、最初の 100 バイトだけが戻されます。

PMArticles ニックネーム:

以下の表では、PMArticles ニックネーム内の列について説明します。F 列は、フェッチ・キーとして指定された列を示しています。フェッチ・キーを使用すると、照会の処理時間を短縮できる場合があります。

表 48. PubMed PMArticles ニックネーム

列名	データ・タイプ	説明	タグ	フェッチ・キー
PMID	VARCHAR(10) NOT NULL	PMArticles ニックネームと子ニックネームとの結合に使用する主キー列	UID	はい
MedlineID	VARCHAR(10)	Medline ID	UID	はい
Owner	VARCHAR(8) NOT NULL	出版物項目の所有者; 値は NCBI によって定義されており、NLM、NASA、PIP、KIE、HSR、HMD、SIS、NOTNLM のいずれか。指定しない場合のデフォルトは NLM。	なし	いいえ
Status	VARCHAR(32) NOT NULL	NCBI によって定義されている出版物状況。可能な値には In-Process、Completed、Out-of-scope、PubMed-not_MEDLINE が含まれる。	なし	いいえ

表 48. PubMed PMArticles ニックネーム (続き)

列名	データ・タイプ	説明	タグ	フェッチ・キー
DateCreated	DATE NOT NULL	出版物項目が作成された日付。	なし	いいえ
DateCompleted	DATE	出版物項目が完成した日付。	なし	いいえ
DateRevised	DATE	出版物項目が改定された日付。	なし	いいえ
ArticleTitle	VARCHAR(250) NOT NULL	記事の表題。	TI	いいえ
Pagination	VARCHAR(32)	記事の完全なページ編集。	なし	いいえ
Abstract	VARCHAR(32000)	記事の要約。	TIAB	いいえ
Affiliation	VARCHAR(250)	第 1 著者の姓と住所	AD	いいえ
AuthorList	VARCHAR(3200)	著者のリスト; 標準化済み	AU	いいえ
LanguageList	VARCHAR(250) NOT NULL	セミコロン区切りリスト	LA	いいえ
PublicationTypeList	VARCHAR(250) NOT NULL	セミコロン区切りリスト	PT	いいえ
VernacularTitle	VARCHAR(250)	日常語による記事の表題。	なし	いいえ
DateOfElectronic Publication	VARCHAR(32)	NCBI スキーマでは、この列の構造は指定されていない	なし	いいえ
Country	VARCHAR(128)	雑誌に記載された資料の国または地域。	なし	いいえ
MedlineTA	VARCHAR(250) NOT NULL	Medline 表題略語。	TA	いいえ
NlmUniqueId	VARCHAR(32)	NlmUniqueID が指定されていない場合には MedlineCode	なし	いいえ
GeneSymbolList	VARCHAR(250)	セミコロン区切りリスト; 1996 年以来使用されていない	なし	いいえ
NumberOfReferences	INTEGER	レビュー記事の文献参照数。	なし	いいえ
PersonalNameSubjectList	VARCHAR(250)	名前のセミコロン区切りリストとして標準化済み	PS	いいえ
KeywordList	VARCHAR(3200)	セミコロン区切りリスト	なし	いいえ
SpaceFlightMissionList	VARCHAR(250)	セミコロン区切りリスト	なし	いいえ

表 48. PubMed PMArticles ニックネーム (続き)

列名	データ・タイプ	説明	タグ	フェッチ・キー
InvestigatorList	VARCHAR(250)	名前のセミコロン区切りリストとして標準化済み	なし	いいえ
PublicationStatus	VARCHAR(32)	資料の状況。	なし	いいえ
ProviderID	VARCHAR(32)	資料の提供者 ID。	なし	いいえ
CitationSubsetList	VARCHAR(250)	セミコロン区切りリスト	SB	いいえ
AllFields	VARCHAR(1)	疑似列; 常に NULL を戻す	ALL	いいえ
TextWords	VARCHAR(1)	疑似列; 常に NULL を戻す	TW	いいえ
PubDate	DATE	雑誌および書籍の出版日付 + medline 日付	DP	いいえ
PubDateString	VARCHAR(32)	雑誌および書籍の出版日付 + medline 日付	DP	いいえ
Title	VARCHAR(250)	書籍または雑誌のタイトル	TA	いいえ
Journal_ISSN	CHAR(9)	雑誌の ISSN。	TA	いいえ
Journal_Volume	VARCHAR(10)	雑誌のボリューム。	VI	いいえ
Journal_Issue	VARCHAR(10)	雑誌の発行。	IP	いいえ
Journal_Coden	VARCHAR(32)	雑誌のコード番号 (図書分類コード)。	なし	いいえ
Journal_ISOAbbreviation	VARCHAR(32)	雑誌の ISO 省略形。	なし	いいえ
Book_Publisher	VARCHAR(128)	書籍の出版社。	なし	いいえ
Book_Authors	VARCHAR(250)	その他の著者のリストとして標準化済み	なし	いいえ
Book_CollectionTitle	VARCHAR(128)	書籍の表題集。	なし	いいえ
Book_Volume	VARCHAR(10)	書籍のボリューム。	なし	いいえ

PMAccession ニックネーム:

以下の表では、PMAccession ニックネーム内の列について説明します。

表 49. PubMed PMAccession ニックネーム

列名	データ・タイプ	説明	タグ
PMID	VARCHAR(10) NOT NULL	PMAccession の子ニックネームと親ニックネームとの結合に使用されるキー。	なし
DataBankName	VARCHAR(250) NOT NULL	データ・バンクの名前。	SI
Accession	VARCHAR(32) NOT NULL	Accession 番号。	SI

PMChemical ニックネーム:

以下の表では、PMChemical ニックネーム内の列について説明します。

表 50. PubMed PMChemical ニックネーム

列名	データ・タイプ	説明	タグ
PMID	VARCHAR(10) NOT NULL	PMChemical の子ニックネームと親ニックネームとの結合に使用されるキー。	なし
NameOfSubstance	VARCHAR(128) NOT NULL	物質の名前。	NM
RegistryNumber	VARCHAR(32) NOT NULL	CAS またはその他の登録番号	RN
CASRegistry	CHAR	Y または N	なし

PMMESHHeading ニックネーム:

以下の表では、PMMESHHeading ニックネーム内の列について説明します。

表 51. PubMed PMMESHHeading ニックネーム

列名	データ・タイプ	説明	タグ
PMID	VARCHAR(10) NOT NULL	PMMESHHeading の子ニックネームと親ニックネームとの結合に使用されるキー。	ID
DescriptorOrName	VARCHAR(128) NOT NULL	MeSH の名前または記述子。	MH ¹
DescriptorIsMajor	CHAR NOT NULL	記述子がメジャーの場合 Y	なし
QualifierOrSubhead	VARCHAR(128)	MeSH の修飾子または副見出し。	SH
QSIIsMajor	CHAR	修飾語または副題がメジャーの場合 Y	なし

注:

1. 照会に述部 "DescriptorIsMajor = Y" が含まれている場合、検索語は MAJR となります。

PMComments ニックネーム:

以下の表では、PMComments ニックネーム内の列について説明します。

表 52. PubMed PMComments ニックネーム

列名	データ・タイプ	説明	タグ
PMID	VARCHAR(10) NOT NULL	PMComments の子ニックネームと親ニックネームとの結合に使用されるキー。	なし
RefSource	VARCHAR(128) NOT NULL	参照のソース。	なし
Type	VARCHAR(32) NOT NULL	CommentOn, CommentIn, ErratumIn, ErratumFor, RepublishedFrom, RepublishedIn, RetractionOf, RetractionIn, UpdateIn, UpdateOf, SummaryForPatents, OriginalReportIn	なし

表 52. PubMed PMComments ニックネーム (続き)

列名	データ・タイプ	説明	タグ
Note	VARCHAR(3200)	注	なし

PMArticleID ニックネーム:

以下の表では、PMArticleID ニックネーム内の列について説明します。

表 53. PubMed PMArticleID ニックネーム

列名	データ・タイプ	説明	タグ
PMID	VARCHAR(10) NOT NULL	PMArticleID の子ニックネームと親ニックネームとの結合に使用されるキー。	なし
ArticleID	VARCHAR(32) NOT NULL	記事の ID。	なし
IdType	VARCHAR(8) NOT NULL	doi, pii, pmcpid, pmpid, sici, pubmed, medline, pmcid	なし

関連資料:

- 228 ページの『スキーマ・データ・エレメントの単純化』
- 226 ページの『カスタム関数と Entrez 照会』
- 235 ページの『Nucleotide スキーマ表』

Nucleotide スキーマ表

Nucleotide スキーマは、Nucleotide タイプのサーバーから得られるデータの表示方法を定義します。このスキーマは、以下の関連する複数のニックネームで構成されています。

- GBSeq
- GBReference
- GBFeatures
- GBIntervals
- GBQualifiers

次の表では、ニックネーム内の列に関する情報がリストされています。「修飾子」列では、列に関する有効な検索修飾子が示されています。有効な探索タグのリストについては、以下の Web サイトを参照して、「Search Field Descriptions and Qualifiers」へのリンクを見つけてください。

www.ncbi.nlm.nih.gov/entrez/query/static/help/Summary_Matrices.html

ニックネームを作成する場合、列に関するデフォルトのデータ・タイプをオーバーライドできます。たとえば、GBSeq ニックネームの「Sequence」列のデフォルト・データ・タイプである VARCHAR(32000) などです。このデータ・タイプを CLOB(1 MB) に変更することができます。Entrez ラッパーは、最大で 5 メガバイトまでの長さの CLOB データ・タイプをサポートしています。

GBSeq:

次の表では、GBSeq ニックネームの列について説明されています。F 列は、フェッチ・キーとして指定された列を示しています。フェッチ・キーを使用すると、照会の処理時間を短縮できる場合があります。

表 54. Nucleotide GBSeq ニックネーム

列名	データ・タイプ	説明	修飾子	フェッチ・キー
PrimaryAccession	VARCHAR(16) NOT NULL	主 ACCESSION 番号	PACC	はい
SequenceKey	VARCHAR(32) NOT NULL	GBSeq ニックネームを子ニックネームに結合するために使用される主キー列。		いいえ
LocusName	VARCHAR(16) NOT NULL	ローカスの名前。	ACCN	いいえ
SeqLength	INTEGER NOT NULL	シーケンスの長さ。	SLEN	いいえ
Strandedness	VARCHAR(32)	not-set, single-stranded, double-stranded, mixed-stranded	なし	いいえ
MoleculeType	VARCHAR(16)	nucleic-acid, dna, rna, trna, rrna, mrna, urna, snrna, snorna, peptide	PROP	いいえ
Topology	VARCHAR(16)	linear, circular	なし	いいえ
Division	CHAR(3) NOT NULL	GenBank 部。	PROP	いいえ
UpdateDate	DATE NOT NULL	最新の更新日付。	MDAT	いいえ
CreateDate	DATE NOT NULL	レコードが作成された日付。	なし	いいえ
Definition	VARCHAR(7000) NOT NULL	シーケンスの定義行。	TITL	いいえ
Version	INTEGER	シーケンスのバージョン ID。	なし	いいえ
GI	VARCHAR(16)	GenInfo (GI) シーケンス ID。	なし	いいえ
KeywordList	VARCHAR(7000)	セミコロン区切りリスト	KYWD	いいえ
Segment	VARCHAR(250)	セグメント。	なし	いいえ
Source	VARCHAR(200) NOT NULL	ソース。	ORGN	いいえ
Organism	VARCHAR(7000) NOT NULL	有機体。	ORGN	いいえ
Taxonomy	VARCHAR(7000) NOT NULL	分類法。	なし	いいえ
Comment	VARCHAR(7000)	コメント	なし	いいえ
Primary	VARCHAR(7000)	主。	なし	いいえ

表 54. Nucleotide GBSeq ニックネーム (続き)

列名	データ・タイプ	説明	修飾子	フェッチ・キー
SourceDB	VARCHAR(250)	ソース・データベース。	なし	いいえ
Sequence	VARCHAR(32000)	シーケンス。	なし	いいえ
AllText	VARCHAR(1)	疑似列、常に NULL を戻す	ALL	いいえ
FreeText	VARCHAR(1)	疑似列、常に NULL を戻す	WORD	いいえ

GBReference:

次の表では、GBReference ニックネームの列について説明されています。

表 55. Nucleotide GBReference ニックネーム

列名	データ・タイプ	説明	修飾子
SequenceKey	VARCHAR(32) NOT NULL	GBReference 子ニックネームをその親ニックネームへ結合するのに使用されるキー。	なし
ReferenceNum	INTEGER NOT NULL	GBReference_reference から構文解析される	なし
RangeLow	INTEGER NOT NULL	参照の下限 (GBReference_reference から構文解析)	なし
RangeHigh	INTEGER NOT NULL	参照の上限 (GBReference_reference から構文解析)	なし
Authors	VARCHAR(3200)	GenBank 形式による氏名のセミコロンの区切りリスト	AUTH
Consortium	VARCHAR(250)	コンソーシアム。	なし
Title	VARCHAR(250)	GenBank 参照タイトル。	WORD
Journal_Title	VARCHAR(250) NOT NULL	ジャーナルのタイトル。	JOUR
MedlineID	INTEGER	Medline ID	なし
PubMedID	INTEGER	PubMed ID	なし
Remarks	VARCHAR(3200)	注釈	なし

GBFeatures:

次の表では、GBFeatures ニックネームの列について説明されています。

表 56. Nucleotide GBFeatures ニックネーム

列名	データ・タイプ	説明	修飾子
SequenceKey	VARCHAR(32) NOT NULL	GBFeatures 子ニックネームをその親ニックネームへ結合するのに使用されるキー。	なし

表 56. Nucleotide GBFeatures ニックネーム (続き)

列名	データ・タイプ	説明	修飾子
FeatureJoinKey	VARCHAR(32) NOT NULL	GBFeatures ニックネームを子ニックネームに結合するために使用される主キー列。	なし
FeatureKey	VARCHAR(20) NOT NULL		FKEY
FeatureLocation	VARCHAR(200) NOT NULL		なし

GBIntervals:

次の表では、GBIntervals ニックネームの列について説明されています。

表 57. Nucleotide GBIntervals ニックネーム

列名	データ・タイプ	説明	修飾子
FeatureJoinKey	VARCHAR(32) NOT NULL	GBIntervals 子ニックネームをその親ニックネームへ結合するのに使用されるキー。	なし
IntervalFrom	INTEGER		なし
IntervalTo	INTEGER		なし
IntervalPoint	INTEGER		なし
IntervalAccession	VARCHAR(32) NOT NULL		なし

GBQualifiers:

次の表では、GBQualifiers ニックネームの列について説明されています。

表 58. Nucleotide GBQualifiers ニックネーム

列名	データ・タイプ	説明	修飾子
FeatureJoinKey	VARCHAR(32) NOT NULL	GBQualifiers 子ニックネームをその親ニックネームへ結合するのに使用されるキー。	なし
QualifierName	VARCHAR(50)	修飾子の名前	なし
QualifierValue	VARCHAR(32000)	修飾子の値	なし

関連資料:

- 228 ページの『スキーマ・データ・エレメントの単純化』
- 230 ページの『PubMed スキーマ表』
- 226 ページの『カスタム関数と Entrez 照会』

Entrez ラッパーのメッセージ

ここでは、Entrez のラッパーを処理するときに出されるメッセージについて説明します。この表にないメッセージについては、「メッセージ・リファレンス 第 1 巻」、または「メッセージ・リファレンス 第 2 巻」を参照するか、または IBM ソフトウェア・サポートにお問い合わせください。

表 59. Entrez のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0142N	SQL ステートメントはサポートされません。	無効な照会タイプがラッパーに渡されました。発行された SQL ステートメントがこのラッパーでサポートされているものであることを確認してください。
SQL0204N	"<name>" は未定義の名前です。	無効な名前が指定されました。CREATE NICKNAME ステートメントをチェックしてください。
SQL0405N	数値リテラル "<literal>" は、値が有効な値の範囲外になっているため、無効です。	検出された XML 内の列、または SQL ステートメントの述部の値は、そのデータ・タイプで可能な範囲に含まれていません。この列とデータ・ソースの列のデータ・タイプを確認してください。または、より適切なタイプでその列を再定義してください。
SQL0408N	値には、その割り当てターゲットのデータ・タイプとの互換性がありません。ターゲット名は "<target_name>" です。	XML データの列に、データ・タイプに無効な文字が入っています。この列とデータ・ソースの列のデータ・タイプを確認してください。または、より適切なタイプでその列を再定義してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「データベースのプロトタイプが見つからない」。)	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「解凍するデータがない」。)	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。

表 59. Entrez のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「ラッパー・オブジェクトの作成エラー」。)	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「式タイプの間違い」。)	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「ニックネームが見つからない」。)	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「メモリー割り振りエラー」。)	ラッパー内で割り振り要求を処理するためのメモリーが足りません。
SQL1816N	ラッパー "<wrapper_name>" は、フェデレーテッド・サーバーに定義を試みているデータ・ソース ("<server_type>", "<server_version>") の "version" にアクセスするために使用できません。	CREATE SERVER ステートメントの VERSION 文節の値が無効です。
SQL1816N	ラッパー "<wrapper_name>" は、フェデレーテッド・サーバーに定義を試みているデータ・ソース ("<server_type>", "<server_version>") の "type" にアクセスするために使用できません。	CREATE SERVER ステートメントの TYPE 文節の値が無効です。
SQL1817N	CREATE SERVER ステートメントは、フェデレーテッド・データベースに定義したデータ・ソースの "type" を示していません。	CREATE SERVER ステートメントの TYPE 文節は必須ですが、指定されていませんでした。

表 59. Entrez のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Entrez Wrapper" から予期しないエラー・コード "900" を受け取りました。関連するテキストおよびトークンは「Parent ニックネームが定義されていない」です。	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1823N	サーバー "<server_name>" からデータ・タイプ "<data_type>" に存在するデータ・タイプ・マッピングがありません。	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1881N	"<option_name>" は、"<option_name>" のための有効な "<option_type>" ではありません。	指定されたオプションは、有効なオプションではありません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1882N	"<option_type>" オプション "<option_name>" は、"<option_name>" に対して "<option_value>" に設定できません。	このオプションに対して指定された値が無効です。CREATE NICKNAME ステートメントをチェックしてください。
SQL1883N	"<option_name>" は、"<option_name>" に必須の "<option_type>" オプションです。	指定されたオプションは、このオブジェクトの場合に必須ですが、指定されていませんでした。CREATE NICKNAME ステートメントをチェックしてください。
SQL1884N	複数の "FOREIGN_KEY" ("COLUMN" オプション) を指定しました。	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1884N	複数の "PRIMARY_KEY" ("COLUMN" オプション) を指定しました。	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「サーバーのバージョンを変更できない」。	ALTER SERVER ステートメントを実行することによってサーバーのバージョンを変更することはできません。新しいバージョンの新しいサーバーを作成する必要があります。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「無効な PARENT ニックネーム」。	PARENT ニックネーム・オプションの中で参照されているニックネームは、現在のニックネームに対して無効です。

表 59. Entrez のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「無効な列名」。	CREATE NICKNAME ステートメントで指定されている列名が、そのニックネームで可能な列のどれにも一致しません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「フェッチ・キーを AND で連結できない」。	1 つの結合の中で、フェッチ・キー (PMArticles ニックネームの PMID 列など) に対する複数の参照がなされました。たとえば、"PMID = 12346 AND PMID = 12348" のようになっています。フェッチ・キー述部は、OR を使用してのみ連結できます。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「SEARCH_TERM と CONTAINS 関数が混在」。	SEARCH_TERM 関数と CONTAINS 関数を 1 つの照会の中で混在させることはできません。1 つの照会では、1 つの SEARCH_TERM 関数しか使用できません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「関数の最初の引き数が無効」。	SEARCH_TERM 関数または CONTAINS 関数の最初の引き数が無効です。この引き数は、列への参照でなければなりません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「関数の 2 番目の引き数が無効」。	SEARCH_TERM 関数または CONTAINS 関数の第 2 の引き数が無効です。この引き数は、ストリング・リテラル、ホスト変数、または列参照でなければなりません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「CONTAINS 関数にタグのない列がある」。	CONTAINS 関数の最初の引き数が無効です。この引き数は、タグ付き列への参照でなければなりません。
SQL30090N	操作がアプリケーション実行環境で無効です。理由コード = 「無効な関数」。	これは内部エラーです。IBM ソフトウェア・サポート担当者に連絡してください。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 12 章 Excel データ・ソースへのアクセスの構成

この章では、Excel データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Excel データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- Excel の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- Excel ラッパーに関連するエラー・メッセージをリストします。

Excel とは?

Excel スプレッドシートまたはワークブックとは、Microsoft® Excel アプリケーションを使用して作成されるファイルで、xls というファイル拡張子を持ちます。DB2® Information Integrator は、Excel 97、Excel 2000、および Excel 2002 のスプレッドシートをサポートします。図 23 は、Excel ラッパーがスプレッドシートとフェデレーテッド・システムを接続する方法を示しています。

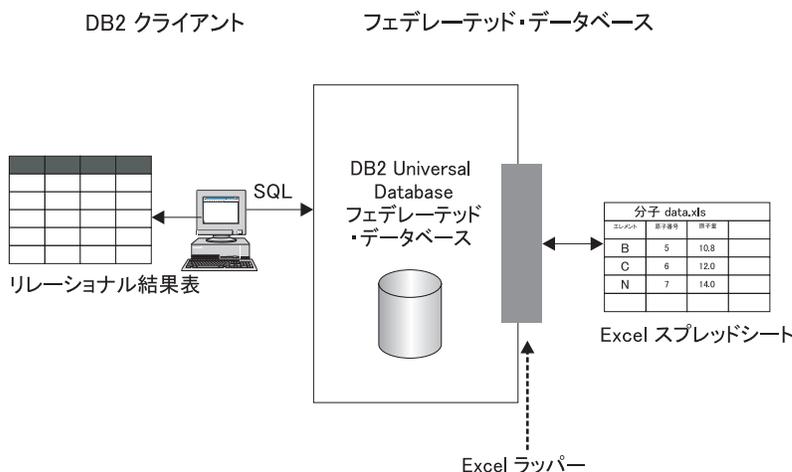


図 23. Excel ラッパーがどのように機能するか

Excel ラッパーは CREATE NICKNAME ステートメントを使用して、Excel スプレッドシート内の列を DB2 UDB システム内の列にマップします。表 60 は、Compound_Master.xls というファイルに保管されるサンプル・スプレッドシート・データを示しています。

表 60. Compound_Master.xls のサンプル・スプレッドシート・データ

	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED

表 60. *Compound_Master.xls* のサンプル・スプレッドシート・データ (続き)

	A	B	C	D
2	compound_A	1.23	367	テスト済み
3	compound_G		210	
4	compound_F	0.000425536	174	テスト済み
5	compound_Y	1.00256		テスト済み
6	compound_Q		1024	
7	compound_B	33.5362		
8	compound_S	0.96723	67	テスト済み
9	compound_O	1.2		テスト済み

通常、この情報は標準 SQL コマンドでは使用できません。Excel ラッパーがインストールされて登録されると、この情報が標準リレーショナル・データ・ソースであるかのようにアクセスすることができます。たとえば、分子量が 100 より大きいすべての化合物データを知りたい場合、以下の SQL 照会を実行します。

```
SELECT * FROM compound_master WHERE mol_count > 100
```

照会の結果が、表 61 に示されています。

表 61. 照会の結果

COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
compound_A	1.23	367	テスト済み
compound_G		210	
compound_F	0.000425536	174	テスト済み
compound_Q		1024	

関連概念:

- 29 ページの『Excel データにアクセスする方法』

関連タスク:

- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』

フェデレーテッド・サーバーへの Excel の追加

フェデレーテッド・サーバーへの Excel データ・ソースの追加

フェデレーテッド・サーバーが Excel データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、フェデレーテッド・サーバーが Excel データ・ソースにアクセスするように構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- ラッパーがデータにアクセスできるように、適切に構造化された Excel ワークシート

手順:

Excel データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. ラッパーを登録します。
2. サーバー定義を登録します。
3. Excel ワークシートのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 29 ページの『Excel データにアクセスする方法』
- 243 ページの『Excel とは?』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 245 ページの『Excel ラッパーの登録』
- 247 ページの『Excel データ・ソースのサーバーの登録』
- 247 ページの『Excel データ・ソースのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』
- 28 ページの『正しいラッパーの選択』

Excel ラッパーの登録

Excel ラッパーの登録は、Excel データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Excel データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

制約事項:

- Excel ラッパーは、DB2 Universal Database Enterprise Server Edition をサポートする Microsoft Windows オペレーティング・システムでのみ使用可能です。
- Excel ラッパーを使用する前に、DB2 Information Integrator がインストールされているサーバーに、Excel アプリケーションをインストールしておく必要があります。
- パススルー・セッションは使用できません。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、ラッパーを excel_wrapper の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER excel_wrapper LIBRARY 'db2lsxls.dll';
```

ラッパー・ライブラリー・ファイル db2lsxls.dll を CREATE WRAPPER ステートメント内に指定する必要があります。

この一連の作業における次のタスクは、『Excel ラッパーのサーバー定義の登録』です。

関連タスク:

- 247 ページの『Excel データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Excel ラッパーのライブラリー・ファイル

次の表では、Excel ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールする場合、表でリストされているディレクトリー・パスに、以下のライブラリー・ファイルが追加されます。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 62. Excel ラッパーのライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
Windows	%DB2PATH%\%bin	db2lsxls.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 245 ページの『Excel ラッパーの登録』

Excel データ・ソースのサーバーの登録

Excel データ・ソースのサーバーの登録は、Excel をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。ラッパーが登録された後、対応するサーバーを登録する必要があります。

Excel の場合は、フェデレーテッド・オブジェクトの階層でデータ・ソース・ファイル (ニックネームで識別される) と特定のサーバー・オブジェクトを関連付ける必要があるため、サーバー定義が作成されます。

手順:

Excel サーバーをフェデレーテッド・システムに登録するには、CREATE SERVER ステートメントを使用します。

biochem_lab という、生化学データを含むワークブックのサーバー・オブジェクトを作成するとします。このサーバー・オブジェクトを、CREATE WRAPPER ステートメントを使用して登録した Excel ラッパー名と関連付けなければなりません。このサーバー・オブジェクトを登録する CREATE SERVER ステートメントは、以下のようになります。

```
CREATE SERVER biochem_lab WRAPPER Excel_2000_Wrapper;
```

この一連の作業における次のタスクは、『Excel データ・ソースのニックネームの登録』です。

関連タスク:

- 247 ページの『Excel データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 597 ページの『CREATE SERVER ステートメントの引き数 - Excel ラッパー』

Excel データ・ソースのニックネームの登録

Excel データ・ソースのニックネームの登録は、Excel をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で Excel データ・ソースを参照する場合に使用されます。

手順:

Excel データ・ソースをリレーショナル表にマップするには、CREATE NICKNAME ステートメントを使用してニックネームを作成します。

以下の例のステートメントは、CompoundMaster.xls という Excel スプレッドシート・ファイルから Compounds ニックネームを作成しています。ファイルには、フェデレーテッド・システムに対して定義される 3 つのデータ列 (Compound_ID、CompoundName、および MolWeight) が含まれています。

```
CREATE NICKNAME Compounds (  
    Compound_ID INTEGER,  
    CompoundName VARCHAR(50),
```

```
MolWeight FLOAT)
FOR SERVER biochem_lab
OPTIONS (FILE_PATH 'C:\My Documents\CompoundMaster.xls',
        RANGE 'B2:D5');
```

一連の作業はこれで終わりです。

関連タスク:

- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 248 ページの『Excel データ・ソース - 照会の例』
- 597 ページの『CREATE NICKNAME ステートメント構文 - Excel ラッパー』

Excel データ・ソース - 照会の例

このトピックでは、ニックネームの例として `Compounds` を使用していくつかのサンプル Excel スプレッドシート照会をリストしています。

照会を実行するには、通常を表名と表の列を使用する場合と同様に、SQL ステートメントでニックネームと定義済みのニックネーム列を使用します。

以下の照会は、分子量が 200 より大きいすべての `compound_ID` を表示します。

```
SELECT compound_ID
FROM Compounds
WHERE MolWeight > 200;
```

以下の照会は、化合物名または分子量が `NULL` になっているすべてのレコードを表示します。

```
SELECT *
FROM Compounds
WHERE CompoundName IS NULL
OR MolWeight IS NULL;
```

以下の照会は、化合物名にストリング `ase` が含まれており、分子量が 300 以上のすべてのレコードを表示します。

```
SELECT *
FROM Compounds
WHERE CompoundName LIKE '%ase%'
AND MolWeight >=300;
```

関連資料:

- 209 ページの『Documentum データ・ソース - 照会の例』
- 249 ページの『Excel データ・ソース - サンプル・シナリオ』

Excel データ・ソース - サンプル・シナリオ

このセクションでは、C:\%Data ディレクトリーにある Excel 2000 ワークシートにアクセスする Excel_2000 ラッパーのサンプルのインプリメンテーションについて説明します。シナリオはラッパー、サーバー、および 1 つのニックネームを登録しますが、それらはワークシートにアクセスするために使用されます。シナリオに示されるステートメントは、DB2 コマンド行を使用して入力されます。ラッパーが登録された後、ワークシートに対して照会ができます。

シナリオは、Compound_Master.xls という化合物ワークシートから始まります。このスプレッドシートは 4 つの列と 9 つの行を持ちます。ファイルの完全修飾パス名は、C:\%Data\%Compound_Master.xls です。内容は、表 63 に示されています。

表 63. サンプル・ワークシート Compound_Master.xls

	A	B	C	D
1	COMPOUND_NAME	WEIGHT	MOL_COUNT	WAS_TESTED
2	compound_A	1.23	367	テスト済み
3	compound_G		210	
4	compound_F	0.000425536	174	テスト済み
5	compound_Y	1.00256		テスト済み
6	compound_Q		1024	
7	compound_B	33.5362		
8	compound_S	0.96723	67	テスト済み
9	compound_O	1.2		テスト済み

手順:

Excel ラッパーを使用して Excel 2000 ワークシートにアクセスするには、以下を行います。

1. Excel_2000 ラッパーを登録します。

```
db2 => CREATE WRAPPER Excel_2000 LIBRARY 'db21sx1s.dll'
```
2. サーバーを登録します。

```
db2 => CREATE SERVER biochem_lab WRAPPER Excel_2000
```
3. Excel ワークシートを参照するニックネームを登録します。

```
db2 => CREATE NICKNAME Compound_Master (compound_name VARCHAR(40),  
weight FLOAT, mol_count INTEGER, was_tested VARCHAR(20))  
FOR biochem_lab  
OPTIONS ( FILE_PATH 'C:\%Data\%Compound_Master.xls')
```

登録プロセスが完了します。これで、Excel データ・ソースはフェデレーテッド・システムの一部となり、SQL 照会で使用できるようになります。

以下の例は、サンプル SQL 照会および Excel データ・ソースを使用して得られる結果を示しています。

- サンプル SQL 照会: 「mol_count が 100 よりも大きい化合物データをすべて取得する」

```
SELECT * FROM compound_master WHERE mol_count > 100
```

結果: 行 2、3、4、6、および 8 のすべてのフィールド。

- サンプル SQL 照会: 「mol_count が判別されていない、すべての化合物の compound_name および mol_count を取得する。」

```
SELECT compound_name, mol_count FROM compound_master  
WHERE mol_count IS NULL
```

結果: ワークシートの行 5、7、および 10 にあるフィールド compound_name および mol_count。

- サンプル SQL 照会: 「未検査で、分子量が 1 よりも大きい化合物の数をカウントする。」

```
SELECT count(*) FROM compound_master  
WHERE was_tested IS NULL AND weight > 1
```

結果: レコード・カウントの 1。これは基準に適合するワークシートの単一行 7 を表す。

- サンプル SQL 照会: 「mol_count が判別されていて、平均 mol_count よりも小さいすべての化合物の compound_name および mol_count を取得する。」

```
SELECT compound_name, mol_count  
FROM compound_master  
WHERE mol_count IS NOT NULL  
AND mol_count < (SELECT AVG(mol_count) FROM compound_master  
WHERE mol_count IS NOT NULL AND was_tested IS NOT NULL)
```

副照会は主照会に平均値 368 を戻し、その後主照会は表 64 を戻します。

表 64. 照会の結果

COMPOUND_NAME	MOL_COUNT
compound_A	367
compound_G	210
compound_F	174
compound_S	67

関連タスク:

- 244 ページの『フェデレーテッド・サーバーへの Excel データ・ソースの追加』

関連資料:

- 248 ページの『Excel データ・ソース - 照会の例』

Excel ラッパーのファイル・アクセス・コントロール・モデル

データベース管理システムは、DB2 データベース・サービスの LOG ON AS プロパティの権限によって Excel ファイルにアクセスします。この設定は、DB2 インスタンスの LOG ON プロパティ・ページに表示できます。プロパティ・ページは、Windows NT の「サービス」コントロール・パネルを介してアクセスします。

関連資料:

- 404 ページの『表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル』

Excel ラッパーのメッセージ

このセクションでは、Excel のラッパーを処理する際に出されるメッセージをリストして説明します。

表 65. Excel のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL1817N	CREATE SERVER ステートメントは、フェデレーテッド・データベースに定義したいデータ・ソースの "VERSION" を示していません。	CREATE SERVER ステートメント作成時に VERSION パラメーターが指定されませんでした。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1000.<internal program code>" を受け取りました。関連するテキストおよびトークンは「メモリー割り振りエラー」です。	IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1001.<internal program code>" を受け取りました。関連するテキストおよびトークンは「不明なオプション」です。	DDL ステートメントで指定したオプションはサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1002.<internal program code>" を受け取りました。関連するテキストおよびトークンは「DELTA オブジェクトの作成が失敗」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1100.<internal program code>" を受け取りました。関連するテキストおよびトークンは「ラッパー・オプションはサポートされていない」です。	ラッパー OPTIONS はこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1200.<internal program code>" を受け取りました。関連するテキストおよびトークンは「<option> はサポートされていない Server オプション」です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。

表 65. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1201.<internal program code>" を受け取りました。関連するテキストおよびトークンは「サーバー名の取得エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1209.<internal program code>" を受け取りました。関連するテキストおよびトークンは「VARCHAR データの変換エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1211.<internal program code>" を受け取りました。関連するテキストおよびトークンは「INTEGER データの変換エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1212.<internal program code>" を受け取りました。関連するテキストおよびトークンは「FLOAT データの変換エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1400.<internal program code>" を受け取りました。関連するテキストおよびトークンは「<option> はサポートされていないユーザー・オプション」です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1401.<internal program code>" を受け取りました。関連するテキストおよびトークンは「USER Delta オブジェクトの作成が失敗」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1500.<internal program code>" を受け取りました。関連するテキストおよびトークンは「<option> はサポートされていないニックネーム・オプション」です。	指定したオプションはこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。

表 65. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1501.<internal program code>" を受け取りました。関連するテキストおよびトークンは「必須オプション PATH が指定されていない」です。	NICKNAME を登録するためには PATH オプションが必要です。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1502.<internal program code>" を受け取りました。関連するテキストおよびトークンは「NICKNAME Delta オブジェクトの作成が失敗」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1503.<internal program code>" を受け取りました。関連するテキストおよびトークンは「Nickname 列タイプの取得エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1504.<internal program code>" を受け取りました。関連するテキストおよびトークンは「Nickname 列タイプ名の取得エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1505.<internal program code>" を受け取りました。関連するテキストおよびトークンは「データ・ソース Excel Wrapper からの受け取り」です。	指定した <data type> はこのラッパーではサポートされません。SQL ステートメントを変更して再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1506.<internal program code>" を受け取りました。関連するテキストおよびトークンは「Nickname 列情報の取得エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1507.<internal program code>" を受け取りました。関連するテキストおよびトークンは「<option> オプションはドロップできない」です。	指定したオプションは必須オプションのため、ドロップできません。

表 65. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1508.VANI" を受け取りました。関連するテキストおよびトークンは「列名は変更できない」です。	列名の変更は Excel ラッパーによって許可されていません。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1509.VCTS" を受け取りました。関連するテキストおよびトークンは「列情報が見つからない」です。	列情報は見つかりません。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1701.<internal program code>" を受け取りました。関連するテキストおよびトークンは「SQL の構文解析エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1702.<internal program code>" を受け取りました。関連するテキストおよびトークンは「NICKNAME オブジェクトのアクセス・エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1703.<internal program code>" を受け取りました。関連するテキストおよびトークンは「データ・ストレージ域の構築エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1704.<internal program code>" を受け取りました。関連するテキストおよびトークンは「SQL から Nickname Data へのリンク・エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

表 65. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1705.<internal program code>" を受け取りました。関連するテキストおよびトークンは「Excel アプリケーションの始動が失敗」です。	Excel アプリケーションの始動が失敗しました。Excel がシステムにインストールされていて、正しいバージョンのラッパーが登録されていることを確認してください。Windows NT の「サービス」コントロール・パネルで、DB2 インスタンスの LOG ON AS プロパティを調べてください。この権限を使用して Excel アプリケーションにアクセスします。このユーザーに適切な権限があることを確認するか、またはこのプロパティを許可アカウントに変更した後、DB2 を再始動して SQL 照会を再実行してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1706.<internal program code>" を受け取りました。関連するテキストおよびトークンは「ソース・スプレッドシートのオープン・エラー」です。	SQL 照会のニックネームによって参照されるスプレッドシートをオープンするときに問題が発生しました。CREATE NICKNAME ステートメント時に指定した PATH にファイルが登録時に存在するようにしてください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1707.<internal program code>" を受け取りました。関連するテキストおよびトークンは「DL 出力ストレージ域のアクセス・エラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1708.<internal program code>" を受け取りました。関連するテキストおよびトークンは「Excel アプリケーションの終了が失敗」です。	内部プログラム・エラーが発生しました。何度照会を繰り返してもこのエラーが続く場合は、IBM ソフトウェア・サポート担当者に連絡してください。
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1711.<internal program code>" を受け取りました。関連するテキストおよびトークンは「取り出し時のエラー、データ/列タイプの不整合の可能性」です。	SQL 照会の際に取り出されたデータは、ニックネームの登録時に指定したデータ・タイプと異なるデータ・タイプでした。ソース・スプレッドシートのデータを変更するか、またはニックネームに登録したデータ・タイプを変更してください。これによって問題が解決しない場合は、IBM ソフトウェア・サポート担当者に連絡してください。

表 65. Excel のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Excel Wrapper" から予期しないエラー・コード "-1900.<internal program code>" を受け取りました。関連するテキストおよびトークンは「メモリー割り振りエラー」です。	内部プログラム・エラーが発生しました。IBM ソフトウェア・サポート担当者に連絡してください。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 13 章 Extended Search データ・ソースへのアクセスの構成

この章では、IBM Lotus Extended Search データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Extended Search データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- Extended Search の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- Extended Search ラッパーに関連するエラー・メッセージをリストします。

Extended Search とは?

Extended Search (拡張検索) 製品は、豊富な検索機能を提供する、複数層からなるクライアント/サーバー・システムです。Extended Search を使用すれば、要求を 1 回入力するだけで、何千ものデータ・リポジトリとインターネットの中を同時に検索することができます。それらのリポジトリとしては、さまざまな内容や構造のものが可能であり、世界中のあちこちに分散して存在するものを同時に指定できます。

Extended Search では、分散/異機種混合の検索により、構造化データや非構造化データを単一のアクセス・ポイントから検索することがサポートされています。既存のデータ管理資産を活用しながら、多種多様な情報源に同時にアクセスするために必要となる複雑な処理を完全に実行します。

Extended Search は、共通の検索構文として独自の汎用照会言語 (GQL) を使用します。各検索要求は、内部で、検索対象のデータ・ソースに適した固有の検索言語に変換されます。さらに、各情報源に固有の方式を使用して探索し、情報源がどこにあるかに依存しない形で情報を取り出します。

Extended Search サーバーのインストール、検索ドメインの構成、および GQL の使用方法については、Extended Search 製品に付属の資料を参照してください。下記の URL にある IBM® Lotus® Extended Search の Web サイトの「Resource」ページでは、その後に示す各資料を参照できます。

<http://www.lotus.com/products/des.nsf/wdocuments/resources>

「Extended Search General Information」

Extended Search システムの各コンポーネントについて、またその相互作用やバックエンド・データ・システムとの相互作用について説明します。

「Extended Search Installation」

製品をインストールするために必要なシステム前提条件やインストールの手順について、およびインストール処理を検証する手順について説明します。

「Extended Search Administration」

検索ドメインにデータ・ソースを追加したり、検索可能フィールドを構成したり、サンプル検索アプリケーションを使用して Extended Search ソースに対する照会を実行したりする手順について説明します。

「Extended Search Programming」

検索サポートを拡張して、製品のデフォルト構成ではサポートされていないさまざまなデータ・ソースも検索可能にするために使うアプリケーション開発ツールについて説明します。Extended Search の汎用照会言語についても、この資料の中で説明されています。

Extended Search のデータ・ソース

Extended Search で検索できるデータ・ソースの種類は、次のものがあります。

- 数多くの一般的な Web 検索サイトおよびニュース・サイト。イントラネット検索サイトやその他の内部/外部検索サイトで検索することが必要になった場合、簡単にそのためのサポートを追加できます。
- メール・システム。 Lotus Notes[®] や Microsoft[®] Exchange Server で管理されているものなど。
- 文書管理システム。 DB2[®] Information Integrator for Content databases など。
- リレーショナル・データベース。 Open Database Connectivity (ODBC) 規格に準拠した IBM DB2、 Oracle、 Microsoft SQL Server、 Microsoft Access などのデータベース。
- フル・テキスト索引。 IBM WebSphere[®] Portal、 Domino[™] Domain Index、 Microsoft Index Server、 Microsoft Site Server で作成したものなど。
- Lotus リポジトリ。これには、ノーツ データベース、 Domino.Doc[®] のライブラリーとキャビネット、 Lotus QuickPlace のプレース、および Lotus Discovery Server[™] のナレッジ・マップ (K-map) が含まれます。
- インスタント・メッセージング・システム (Lotus Sametime[®] など)。この機能を使えば、検索可能なデータ・リポジトリに対して照会を実行するだけでなく、そのことについて知識のある人に直接尋ねることが可能になります。
- Lightweight Directory Access Protocol (LDAP) ディレクトリー。 IBM SecureWay[®]、 Domino LDAP Server、および Exchange LDAP Server で管理するものなど。
- ファイル・システム。ローカルに、またはネットワーク・ドライブに存在するテキスト・ファイルを検索できます。圧縮されたファイルや暗号化されたファイルは検索できません。

Extended Search の C++ および Java[™] アプリケーション・プログラミング・インターフェース (API) を使用すれば、ここに示されていない独自データベースなど、その他の情報源から検索するようにサポートを拡張できます。

Extended Search ラッパーがどのように機能するか

構造化リレーショナル・データベース・モデルにおいて、列は、ある一貫した形式で名前が付けられ、表示されます。それにより、細かい計算処理を実行したり、特定の列の値を比較することによって異なる表のデータを結合したりできます。ま

た、ある表には存在し、別の表には存在しないオブジェクトのリストを出力するなど、さまざまな分析処理を実行できます。

一方、構造化されていないデータは、多くの場合、フリー・テキストとして保管されています。一般に、列名によって情報を照会するために利用できるメタデータは、ほとんど、またはまったくありません。非構造化データを検索する場合には、計算処理による基準に基づいて検索するよりも、ユーザーの指定するキーワードに一致するデータを見つけることによって検索するほうが現実的です。

Extended Search ラッパーは、それら 2 種類の検索テクニックを組み合わせたものです。Extended Search ラッパーを使うと、SQL を使用することによって、Extended Search ドメイン内の非構造化コンテンツ内を検索できます。そして、その検索結果に対して分析処理や関係演算処理を実行できます。

照会を発行するには、特殊な用途の DB2 表 (ニックネーム表) を参照する SQL ステートメントを入力します。Extended Search は、その SQL 基準に基づいて検索を実行し、その結果データをニックネーム表に入れます。検索結果は 1 つの表の中に保たれているため、そのデータを他のデータベース表 (ニックネーム表を含む) の関係する処理に使用することができます。

このラッパーによる検索要求をサブミットすると、ニックネーム表にマップされている任意の Extended Search ソースからデータを取り出すことができます。そのデータは、ネイティブのデータ・ソースの外に移動することなく、フェデレーテッド・システム内の他のデータ・ソースと統合できます。照会に対してどんなに多くの情報源から応答が戻されたとしても、検索結果は 1 つの結果セットとして戻されます。

次の図は、Extended Search ラッパーによって、Extended Search ドメイン内のさまざまなデータ・ソースがフェデレーテッド・データベース・システムに接続される様子を示すものです。ラッパーは、1 つ以上のリモート Extended Search サーバーにあるデータにアクセスし、そこからデータを取り出します。他の Extended Search サーバーに接続されている Extended Search サーバーにラッパーが接続した場合、複数のサーバーから検索結果が戻されることがあります。

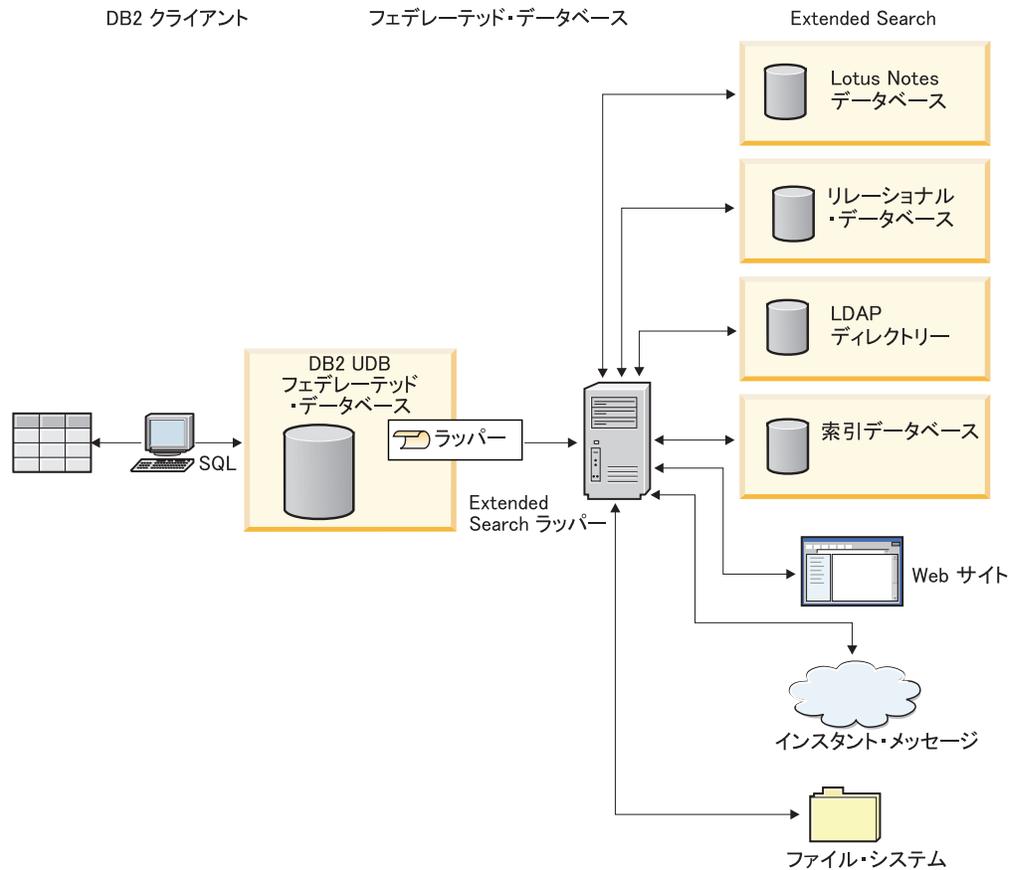


図 24. Extended Search ラッパーがどのように機能するか

関連タスク:

- 264 ページの『フェデレーテッド・サーバーへの Extended Search データ・ソースの追加』

Extended Search のニックネーム

Extended Search のデータ・モデルにおいて、1 つの文書は 1 つ以上のフィールドで構成されます。複数の文書を集めたものが、1 つのデータ・ソースを構成します。また、任意の数のデータ・ソースを組み合わせて 1 つのカテゴリにすることができます。それにより、1 つのまとまりとしてそれらを検索および管理できます。

ユーザーのアクセスを、実際に必要とするデータ・ソースだけに制限するため、カテゴリは少なくとも 1 つのアプリケーションに属していなければなりません。アプリケーションを、アクセスや検索機能を制御するためにユーザーをグループ化する一手段と考えてください。たとえば、人事関係のアプリケーションと財務関係のアプリケーションとでは、同じデータ・ソースが組み込まれているかもしれませんが、それらのデータ・ソースのうち、それぞれのアプリケーションのユーザーのアクセスするフィールドは必ずしも同じではありません。

ニックネームを登録する際には、検索対象となるアプリケーション、カテゴリ、データ・ソース、およびデータ・ソース・フィールドを識別します。それらのエン

ティティは、Extended Search 構成データベース内に存在していなければなりません。Extended Search ラッパーを使って Extended Search データ・ソース内を検索するには、そのデータ・ソースのニックネームを作成する必要があります。

ニックネーム表の内容は、そのニックネームを登録した時点での Extended Search 構成データベースの状態を反映するものです。Extended Search の管理者が構成を更新すると (例: データ・ソースやフィールドを追加または削除した場合)、その変更内容は、ニックネーム表には反映されていません。変更されたデータがニックネーム表の中で参照されている場合、Extended Search 構成データベースを最新に保つには、ニックネームを変更するか、またはそれをドロップしてから新しいニックネームを作成する必要があります。

ニックネームを変更したり再作成したりしないなら、Extended Search ドメインにもはや存在しない項目を検索しようとして結果が 0 であったことを示すエラーとレポートを受け取ることになります。

Extended Search で構成されているすべてのデータ・ソースに関する情報を 1 つのニックネーム表に入れることは可能ですが、複数のニックネーム表を作成したほうが便利な場合もあります。DB2® をフルに活用するには、Extended Search ラッパーで検索するデータ・ソースの種類ごとに、1つずつ別個のニックネームを作成してください。

たとえば、Web ソース用に 1 つのニックネーム、Notes データベース用に 1 つのニックネーム、ファイル・システム用に 1 つのニックネームを作成するとよいかもしれません。複数の別個のニックネーム表を作成することにより、ラッパーに戻されるデータに対して結合操作を実行したり、フィールド値に基づいてさまざまなデータ・ソースを関連付けたり、結果データをフェデレーテッド・システム内の他のデータと統合したりする操作が、さらにうまく実行できるようになります。

関連概念:

- 261 ページの『Extended Search の垂直表』

関連タスク:

- 268 ページの『Extended Search データ・ソースのニックネームの登録』

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 601 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』

Extended Search の垂直表

1 つの Extended Search アプリケーションに多数のカテゴリを含めることができ、その各カテゴリに多数のデータ・ソースを含めることができます。各データ・ソースごとにフィールド名に独自の規則が使用されるため、フィールド名が重複していると、空セットが生成されることがあります。データ・ソース・フィールドをニックネーム表のユーザー定義列にマップし、検索結果を水平表の形で表示した場合、その表の列数が手に負えないほど多くなってしまう可能性があります。ごく一部の列だけにデータが含まれる行が大半を占める場合、その表のデータは非常

にまばらなものになります。例:

Column_1	Column_2	Column_3
Value_11		
	Value_22	
Value_31		Value_33

Extended Search では、マップ・フィールドを定義することによって結果の表示方法を制御できます。マップ・フィールドは、用途は同じだが異なるデータ・ソースで異なる名前の付けられた内容を組み合わせるための手段となるものです。たとえば、さまざまなデータ・ソースの EmpNum、EmpNo、および EmpID の各フィールドから得られる結果データを表示するために、EmployeeNumber という名前のマップ・フィールドを作成することができるでしょう。このようなマッピング機能を使用しない場合は、マップ・フィールドとして 1 つの列を作成するのではなく、固有なフィールド名ごとに 1 つずつニックネーム列を定義することが必要になります。

フィールドのマッピングは、関連付けるフィールドの名前がわかっている場合に便利です。しかし、アプリケーションの中には、多数のデータ・ソースから取られる大量のフィールドを関連付けることが必要なものがあります。特に非構造化データの場合は、それらのフィールド相互の関係を前もって知ることができないかもしれません。したがって、意味のあるニックネーム表を定義し、構造を指定することはとても困難です。このようなアプリケーションをサポートするため、Extended Search ラッパーでは、垂直ニックネーム表を作成する機能が用意されています。

Extended Search のためのニックネーム表を作成する際に、VERTICAL_TABLE オプションを有効にすることができます。このオプションでは、Extended Search 構成データベース内でデータ・ソースのうち戻すことのできるフィールドとして構成されているすべてのフィールドが戻されます。検索にどの列が関係しているかわからない場合や、後処理照会を実行したり結果セットに対して結合を実行したりする際にどの列が関係しているかわからない場合には、このオプションを使用します。

垂直表の各行には、結果セットの中に戻されたフィールドに関する情報が入れられます。行ごとに、フィールドの属するデータ・ソースの名前、フィールド名、その値、データ・タイプ (日付、整数など) が戻されます。水平表の場合には複数の列の間でデータがまばらに点在した結果になりますが、垂直表の場合はデータがびっしりと並び、数多くのデータ行が含まれることになります。例:

Field_Name	Field_Value	Field_Datatype
Column_1	Value_11	VARCHAR
Column_2	Value_22	DATE
Column_1	Value_31	VARCHAR
Column_3	Value_33	VARCHAR

この表に対する照会を実行する際、このデータに対する SQL 操作が可能であり、すべての列ラベルを照会できます。例:

```
Field_Value LIKE '%IBM%'
```

VERTICAL_TABLE オプションはデータ・ソースのうち可能性のあるすべてのフィールドに関する情報を戻すため、特定のユーザー定義列を照会する必要はないかもしれませんが。このオプションを使用可能にして、SELECT ステートメントによりユーザー定義列を検索すると、受け取る検索結果の中に重複した情報が含まれる場合があります。しかし、ユーザー定義列を定義する場合、フェデレーテッド・システム内の他の表との結合操作でそれらの列を使用することができます。

次の 2 つの表は、Extended Search において、垂直ニックネーム表の各行ごとに戻されるシステム提供の列をまとめたものです。

ラッパーはニックネームごとに以下の 3 つの固定列を常に戻します。

列名	データ・タイプ	説明
DOC_ID	VARCHAR(512)	文書 ID。検索結果セット内の項目ごとに固有。
DOC_RANK	INTEGER	その文書の関連性ランキング。
CLIENT_LOCALE	VARCHAR(5)	検索要求のクライアント・ロケール。SQL 照会でクライアント・ロケールが指定されていない場合、デフォルトのクライアント・ロケールとして enUS が使用されます。

次の固定列は、VERTICAL_TABLE オプションが使用可能になっている場合にのみ、ラッパーによって作成されます。

列名	データ・タイプ	説明
DATASOURCE_NAME	VARCHAR(128)	検索結果を生成したデータ・ソースの名前。
FIELD_NAME	VARCHAR(128)	検索結果の中に戻されたフィールドの名前。
FIELD_VALUE	VARCHAR(4096)	結果セットの中に戻されたフィールドの値。そのフィールド値がニックネーム列 (VARCHAR 値) の最大長より長い場合、フィールド値は切り捨てられます。その列の末尾に ES_TRUNCATE というトークンが付加されて、その値が不完全であることが示されます。
FIELD_DATATYPE	SMALLINT	フィールド値の実際のデータ・タイプを表す整数値。その値は次のとおり。 384 DATE 448 VARCHAR 496 INTEGER

結果データを VARCHAR 値として格納する垂直表は、照会が困難な場合があります。検索をもっと絞り込むには、Extended Search 構成データベース内にマップ・フィールドを作成し、ニックネーム表の中でそれらを定義してください。マップ・フィールドを使用すれば、検索結果のコンパクトな水平表を作成できます。また、結果に対して関係演算を実行したり、フェデレーテッド・データベース・システム内の他の表も関係する照会の中でそれらを組み合わせたりするように最適化することもできます。

Extended Search のマップ・フィールドの定義方法については、「*Extended Search Administration*」を参照してください。これは、IBM® Lotus® Extended Search の Web サイトで入手できます。その URL は次のとおりです。

<http://www.lotus.com/products/des.nsf/wdocuments/resources>

関連概念:

- 260 ページの『Extended Search のニックネーム』

関連タスク:

- 268 ページの『Extended Search データ・ソースのニックネームの登録』

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 601 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』

フェデレーテッド・サーバーへの Extended Search の追加

フェデレーテッド・サーバーへの Extended Search データ・ソースの追加

Extended Search データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

DB2 コントロール・センターまたは DB2 コマンド行を使用して、Extended Search データ・ソースへアクセスするようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- Extended Search ラッパーを使用する前に、検索を計画しているソースを Extended Search 構成データベース内に構成してください。Extended Search ラッパーを使用してソースを検索する前に、Extended Search クライアントから少数の照会をサブミットして、ソースの検索が可能であることを確認します。

手順:

Extended Search データ・ソースをフェデレーテッド・サーバーに追加するには、次のようにします。

1. ラッパーを登録します。
2. サーバー定義を登録します。
3. オプション: ユーザー・マッピングを作成します。
4. Extended Search データ・ソースのニックネームを登録します。

5. オプション: Extended Search ラッパーのカスタム関数を登録します。

関連概念:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator のインストール - 概要』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 265 ページの『Extended Search ラッパーの登録』
- 266 ページの『Extended Search データ・ソースのサーバーの登録』
- 268 ページの『Extended Search データ・ソースのニックネームの登録』
- 267 ページの『Extended Search データ・ソースのユーザー・マッピングの登録』
- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Extended Search ラッパーの登録

Extended Search ラッパーの登録は、Extended Search データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Extended Search データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

たとえば、Windows オペレーティング・システムを使用するフェデレーテッド・サーバーで、NotesDBwrapper という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER NotesDBwrapper LIBRARY 'db2uies.dll'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『Extended Search ラッパーのライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Extended Search ラッパーのサーバー定義の登録』です。

関連タスク:

- 266 ページの『Extended Search データ・ソースのサーバーの登録』

関連資料:

- 266 ページの『Extended Search ラッパーのライブラリー・ファイル』
- 599 ページの『CREATE WRAPPER ステートメント構文 - Extended Search ラッパー』

Extended Search ラッパーのライブラリー・ファイル

次の表では、Extended Search ラッパーのディレクトリー・パスおよびライブラリー・ファイル名がリストされています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で実行されている場合、ディレクトリー・パスに追加されるラッパーのライブラリー・ファイルは、libdb2uies.a、libdb2uiesF.a、および libdb2uiesU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 66. Extended Search ラッパーのライブラリーのロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2uies.a
Linux	/opt/IBM/db2/V8.1/lib	libdb2uies.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2uies.so
Windows	%DB2PATH%¥bin	db2uies.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:¥Program Files¥IBM¥SQLLIB です。

関連タスク:

- 265 ページの『Extended Search ラッパーの登録』

Extended Search データ・ソースのサーバーの登録

この作業は、Extended Search データ・ソースをフェデレーテッド・システムに追加するためのメインタスクの一部です。ラッパーの登録後、対応するサーバー定義を作成して、フェデレーテッド・システムに統合するリモート Extended Search サーバーを識別する必要があります。この定義によって、ラッパーは Extended Search サーバーに接続できます。

手順:

Extended Search サーバーを登録するには、DB2 コマンド行プロセッサから CREATE SERVER ステートメントを発行します。

たとえば、myESwrapper という名前のラッパー用に es1 という名前のサーバーを登録するには、次のステートメントを発行します。この Extended Search サーバーは、デフォルトのポート値を使用します。

```
CREATE SERVER es1 WRAPPER myESwrapper OPTIONS (ES_HOST 'my.server.com')
```

同じサーバーを作成し、すべてのメッセージ・レベル (重大、非重大、警告、および情報) のトレースを可能にして、そのトレース・メッセージを wrapper ディレクトリーの es1wrapper.log に書き込むには、次のステートメントを発行します。

```
CREATE SERVER es1 WRAPPER myESwrapper OPTIONS (ES_HOST 'my.server.com',  
ES_TRACING 'ON', ES_TRACELEVEL 'CNWI',  
ESTRACEFILENAME '/wrapper/es1wrapper.log')
```

次の作業は、『Extended Search データ・ソースのニックネームの登録』です。

関連タスク:

- 268 ページの『Extended Search データ・ソースのニックネームの登録』

関連資料:

- 599 ページの『CREATE SERVER ステートメント構文 - Extended Search ラッパー』

Extended Search データ・ソースのユーザー・マッピングの登録

この作業は、Extended Search データ・ソースをフェデレーテッド・システムに追加するためのメインタスクにおけるオプションのステップです。

ユーザー・マッピングは、Extended Search ラッパーを使用して Extended Search ソースを照会するユーザーのアクセスを認証する方法を提供します。ユーザーが SQL 照会を登録済みの Extended Search ニックネームに対してサブミットする場合、そのユーザーに対してユーザー・マッピングが定義されていないと、Extended Search ラッパーはデフォルトのユーザー ID とパスワードを使用して、リモート Extended Search サーバーからデータを取り出そうとします。照会しているデータ・ソースが認証を必要とする場合は、空の結果セットが戻される可能性があります。

正しいユーザー ID とパスワードが Extended Search サーバーに渡されるようにするには、Extended Search ソースの検索を許可されたユーザーのユーザー・マッピングをフェデレーテッド・システムに作成します。ユーザー・マッピングを作成するとき、パスワードは DB2 カタログ表に暗号化された形式で格納されます。パスワードは、Extended Search によって DB2 から検索されているソースに渡された際のセキュアなフォーマットを保持します。

Extended Search 構成データベースのセキュリティー設定により、そのユーザー ID とパスワードが検索されているソースへのアクセスを許可されているかどうか、およびそのユーザー ID の追加のマッピングが実行されるかどうかを判別します。

手順:

Extended Search のユーザー・マッピングを登録するには、DB2 コマンド行プロセッサから CREATE USER MAPPING ステートメントを発行します。

このステートメントではマップする必要のある DB2 ユーザー ID、ターゲット・データ・ソースをホストする Extended Search サーバー、およびユーザーがそうしたデータ・ソースにアクセスできるようにするユーザー ID とパスワードを識別する必要があります。

たとえば、以下のステートメントではユーザー ID user1 を登録し、Extended Search サーバー es1 を使用してリモート・データベースを検索できるようにします。

```
CREATE USER MAPPING FOR user1 SERVER es1 OPTIONS
  (REMOTE_AUTHID 'ESUserId', REMOTE_PASSWORD 'abc123def')
```

次の作業は、『Extended Search カスタム関数テンプレートの登録』です。

関連タスク:

- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 601 ページの『CREATE USER MAPPING ステートメント構文 - Extended Search ラッパー』

Extended Search データ・ソースのニックネームの登録

この作業は、Extended Search データ・ソースをフェデレーテッド・システムに追加するためのメインタスクの一部です。サーバーを登録した後、少なくとも 1 つのニックネームを登録する必要があります。ニックネーム表は、Extended Search ドメイン内にある 1 つ以上の検索可能なソースを識別する仮想 DB2 表です。照会をサブミットする際、検索したいソースのニックネームを指定します。

前提条件:

ニックネームを作成する Extended Search サーバーが稼働していることを確認してください。ニックネームを作成するとき、システムは検索する予定のソースとフィールドに関する情報が Extended Search 構成データベースに存在するかどうかを検査します。

手順:

Extended Search ニックネームを登録するには、DB2 コマンド行プロセッサから CREATE NICKNAME ステートメントを発行します。

たとえば、以下のステートメントを発行して、es1 Extended Search サーバーによってホストされる Demo アプリケーションの Web カテゴリに属するすべてのデータ・ソースを検索するためのニックネーム表を作成します。WebTitle および WebDescription フィールドを戻し、デフォルトの検索処理オプションを使用します。

```
CREATE NICKNAME allweb (WebTitle VARCHAR(255), WebDescription VARCHAR(1000))
  FOR SERVER es1 OPTIONS(APPLICATIONID 'Demo', CATEGORY 'Web')
```

次のステートメントを発行して、Science アプリケーションで複数のデータ・ソースを検索するためのニックネーム表を作成します。検索結果を列名の垂直リストとし

て提供し、タイムアウト値を 60 秒に設定します。また、それぞれのソースが最高 100 までの結果文書を戻せるようにし、結果のサイズを 1000 項目に拡張して、結果を作成者名でソートします。

```
CREATE NICKNAME stars (Title VARCHAR(80), Author VARCHAR(40),  
    Abstract VARCHAR(200))  
    FOR SERVER es1 OPTIONS (APPLICATIONID 'Science',  
    DATASOURCES 'Astronomy;NASA Library;Astrophysics', VERTICAL_TABLE 'yes',  
    TIMEOUT '60', MAXHITS '100', TOTALMAXHITS '1000', SORTFIELD 'Author')
```

次の作業は、『Extended Search ラッパーのユーザー・マッピングの登録』です。

関連概念:

- 260 ページの『Extended Search のニックネーム』
- 261 ページの『Extended Search の垂直表』

関連タスク:

- 264 ページの『フェデレーテッド・サーバーへの Extended Search データ・ソースの追加』
- 267 ページの『Extended Search データ・ソースのユーザー・マッピングの登録』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 601 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』

Extended Search ラッパーのカスタム関数の登録

この作業は、Extended Search データ・ソースをフェデレーテッド・システムに追加するためのメインタスクにおけるオプションのステップです。

カスタム関数には、実行可能コードは含まれません。関数を登録すると、照会の中でそれを参照してデフォルトの検索動作を変更できます。Extended Search ラッパーのカスタム関数では、精細な検索式を指定し、ニックネーム表で列として定義されていない内容を検索することが可能です。

制約事項:

- Extended Search 関数は、WHERE 文節でのみ呼び出し可能です。
- WHERE 文節には、検索述部として動作する述部 (Extended Search 関数、または「列名 演算子 定数」タイプの述部) が少なくとも 1 つ含まれている必要があります。
- Extended Search 関数は、スカラー関数テンプレートです。EQUAL (=) 演算子を使用する必要があり、比較値は 1 でなければなりません。
- Extended Search 関数の最初のパラメーターは、関数が適用されるニックネームを識別するアンカー値として動作します。これには検索の結果内での文書のランク (DOC_RANK) などがあります。このパラメーターには INTEGER フィールドを指定してください。このパラメーター (評価はされません) は、SQL 照会に複数のニックネームや、ニックネームと表の組み合わせが含まれる場合に、特に重要です。例:

```
SELECT * FROM es_nickname1, es_nickname2
WHERE eswrapper.es_search(es_nickname1.DOC_RANK, '"IBM"') = 1 AND
      eswrapper.es_search(es_nickname2.DOC_RANK, '"IBM"') = 1
```

手順:

Extended Search カスタム関数を登録するには、以下の CREATE FUNCTION ステートメントを発行します。

```
CREATE FUNCTION eswrapper.es_search(integer, varchar(1024))
  RETURNS INTEGER AS TEMPLATE
  DETERMINISTIC NO EXTERNAL ACTION;
```

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 274 ページの『Extended Search ラッパー - 汎用照会言語』
- 604 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 270 ページの『Extended Search ラッパー - 照会のガイドライン』

Extended Search データ・ソースの照会

Extended Search ラッパー - 照会のガイドライン

Extended Search ラッパーは特定のフォーマットの照会を必要とし、正確な言語基準を満たさない照会をサポートしていません。このトピックでは、照会を作成するガイドラインと、正確な照会構文および誤った照会構文の例を提供します。

多言語の Web ソースの照会

Extended Search が Web ソースにリンクするのに使用するサード・パーティー・ソフトウェアは、ISO-8859-1 コード・ページを使用する言語 (英語、フランス語、ドイツ語、ポルトガル語、およびスウェーデン語など) をサポートしています。そのため、Web ソースを検索する場合、韓国語などの 2 バイト文字セット言語、ヘブライ語などの双方向言語、または他の ISO-8859-1 ではない言語を検索することはできません。検索結果を処理するパーサーは、正しくない文字コードと見なされるものを検出すると失敗します。

CLIENT_LOCALE 値の指定

WHERE 文節に CLIENT_LOCALE 列を含めてクライアント・ロケールの値を設定する場合、AND 述部を使用して検索基準を指定する必要があります。

CLIENT_LOCALE 列で OR 述部を使用することはできません。

例 — 正しい構文

以下に示すのは CLIENT_LOCALE 列を WHERE 文節に含める正しい方法の例です。

```
WHERE CLIENT_LOCALE = 'enUS' AND
      ESWRAPPER.ES_SEARCH(DOC_RANK, '"IBM"')=1
```

```
WHERE ESWRAPPER.ES_SEARCH(DOC_RANK, '"IBM"')=1
AND CLIENT_LOCALE = 'enUS'
```

例 — 正しくない構文

以下の例は、CLIENT_LOCALE 列で OR 述部を使用しようとしているため、正しくありません。

```
WHERE CLIENT_LOCALE = 'enUS' OR  
ESWRAPPER.ES_SEARCH(DOC_RANK, '"IBM"')=1
```

```
WHERE ESWRAPPER.ES_SEARCH(DOC_RANK, '"IBM"')=1  
OR CLIENT_LOCALE = 'enUS'
```

Extended Search 固定列での述部の指定

Extended Search ニックネームを含む SQL ステートメントは、WHERE 文節でニックネーム用の述部を指定しなければなりません。ただし、Extended Search 固定列にある述部は述部と見なされません。

例 — 正しくない構文

以下に示す例は、述部を含んでいないため、正しくない照会です。

```
SELECT * FROM ES_NICKNAME
```

以下に示す例は、述部のみが固定列にあるため、正しくない照会です。

```
SELECT * FROM ES_NICKNAME WHERE DOC_RANK < 20
```

アンバインドされた述部の指定

ユーザー定義列の述部は、述部の値が定数である場合にのみ、Extended Search ラッパーによって処理されます。述部の値がアンバインドされている場合、述部は DB2 エンジンによって処理されます。アンバインドされた述部が SQL ステートメントにある唯一の述部である場合、結果はエラーとなります。Extended Search ニックネームには、Extended Search ラッパーが処理できる述部が必要です。

例 — 正しい構文

以下に示す WHERE ステートメントは、Extended Search ラッパーが処理する述部の例です。

```
SELECT *  
FROM ES_NICKNAME  
WHERE Author = 'Ernest Hemingway'
```

例 — 正しくない構文

以下に示す WHERE ステートメントは、DB2 が処理する述部の例です。

```
SELECT *  
FROM ES_NICKNAME_1, ES_NICKNAME_2  
WHERE ES_NICKNAME_1.Author = ES_NICKNAME_2.Author
```

OR 述部による照会の結合

Extended Search ラッパーは、単純な OR 述部によって結合された、異なるニックネーム表、またはニックネーム表とデータベース表を検索することはできません。同じニックネーム内でのみ OR 述部を使用できます。

例 — 正しくない構文

```
SELECT *
FROM ES_Nickname as N1, TABLE as T1
WHERE N1.Column1 = 'abc' OR T1.Column1 = 'abc'
```

```
SELECT *
FROM ES_Nickname_1 as N1, ES_Nickname_2 as N2
WHERE N1.USerdefCol = 'abc' OR N2.USerdefCol = 'cdf'
```

```
SELECT *
FROM ES_Nickname_1 as N1, ES_Nickname_2 as N2
WHERE ESWRAPPER.ES_SEARCH(N1.DOC_RANK, '"IBM"')=1 OR
      ESWRAPPER.ES_SEARCH(N2.DOC_RANK, '"LOTUS"')=1
```

関連タスク:

- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 274 ページの『Extended Search ラッパー - 汎用照会言語』

Extended Search ラッパー - 照会の例

Extended Search ラッパーで照会を実行するには、典型的な DB2 テーブル名およびテーブル列を指定するのと同様に、登録されたニックネームおよびニックネーム列を SQL ステートメントに指定します。

ここで取り上げるサンプル検索のシナリオは、ある病院チームが最新の医学研究を検索し比較する必要があるというものです。幅広いさまざまな情報源を検索するために、病院は Extended Search サーバーを使用します。Extended Search ドメインには、MedResearch というアプリケーションと、ドキュメント・ベースのデータベース、メール・サーバー、および Web を検索するよう構成されているいくつかのカテゴリが含まれています。

検索することに加え、チームはさまざまな検索結果を比較する必要もあります。たとえば、ある時間フレーム内に記事を発行した人々、薬草やビタミン剤を最近購入した人々、代替医療について E メールで同僚と検討した人々、医療ライセンスの更新を申請した人々を識別する必要があります。構造化検索のために非構造化 Extended Search データを DB2 に組み込む機能のある Extended Search ラッパーにより、ソリューションが提供されます。

病院チームは 3 つのニックネーム、つまり文書リポジトリ用のもの、E メール・システム検索用のもの、そして特定の Web ソース検索用のものを作成することを決定します。Owner および Date フィールドが Extended Search 構成データベースでマップ・フィールドとして定義されるため、ネイティブ・データ・ソースで命名されている方法にかかわらず、結合に使用できます。

文書ニックネーム

```
CREATE NICKNAME MedDocs ( Owner   VARCHAR(80),
                          Date    DATE,
                          Title   VARCHAR(80),
                          Abstract VARCHAR(200) )
FOR SERVER esServer OPTIONS ( APPLICATIONID 'MedResearch',
```

```

CATEGORY 'AMA Library;Medical Records;Pharmacy',
VERTICAL_TABLE 'YES',
TIMEOUT '60', MAXHITS '100',
TOTALMAXHITS '1000' )

```

E メール・ニックネーム

```

CREATE NICKNAME MedMail ( Owner  VARCHAR(80),
                          To      VARCHAR(80),
                          Date    DATE,
                          Subject VARCHAR(80) )
FOR SERVER esServer OPTIONS ( APPLICATIONID 'MedResearch',
                              CATEGORY 'Exchange Server;Lotus Notes',
                              VERTICAL_TABLE 'YES', )
                              TIMEOUT '60', MAXHITS '100',
                              TOTALMAXHITS '1000' )

```

Web ニックネーム

```

CREATE NICKNAME MedWeb ( WebTitle VARCHAR(255),
                        WebDescription VARCHAR(1000) )
FOR SERVER esServer OPTIONS ( APPLICATIONID 'MedResearch',
                              DATASOURCES 'Google!;Alta Vista;CNN',
                              TOTALMAXHITS '500' )

```

次の照会は、表題に Artificial Liver という句を、文書内容に MARS という略語を含む文書の検索です。この結果セットでは、2001 年より前に発行された文書はすべて除外されます。

```

SELECT OWNER, DOC_CONTENT
FROM MedDocs
WHERE ESWRAPPER.ES_Search(DOC_RANK, '( ( TOKEN:EXACT "MARS") AND
                                  ( "TITLE" IN "Artificial Liver") AND
                                  ( "DATE" >= "01/01/2001" ) ) ') = 1

```

次の照会は、代替医療 (alternative medicine) について検討する過去数か月に書かれた Eメールの検索です。

```

SELECT *
FROM MedMail
WHERE ESWRAPPER.ES_Search(DOC_RANK, '(
                              ("SUBJECT" IN "alternative medicine") AND
                              ("DATE" BETWEEN "03/01/2002" AND
                              "09/30/2002" ) ) ') = 1

```

次の照会は、相補的代替医療 (CAM) による療法とその療法がアメリカ社会で認められていることについて言及する Web ソースの検索です。

```

SELECT WebTitle, WebDescription
FROM MedWeb
WHERE ESWRAPPER.ES_Search(DOC_RANK, '(
                              TOKEN:EXACT "CAM therapy" ) AND
                              ( TOKEN:FUZZY "United States" ) ') = 1

```

次の照会は、最近ライセンスを取得した医者で病院の薬局で大量の薬草やビタミン剤を購入した人の検索です。次にこの照会は、それらの医者と代替療法についての Eメールを書いた人との名前を突き合わせます。

```

SELECT N2.OWNER, N2.DATE
FROM MedDocs as N1,
MedMail as N2
WHERE ESWRAPPER.ES_SEARCH(N1.DOC_RANK, '(
                              ("LICENSE_DATE" >= "01/01/2002") AND
                              ( ( ( "PRODUCT" = "HERB") OR ("PRODUCT" = "VITAMIN") ) AND

```

```
        ("QUANTITY" > "1000") ) ) ' ) = 1
AND ESWRAPPER.ES_SEARCH(N2.DOC_RANK, ' ("SUBJECT" IN
                                "alternative medicine") ' ) = 1
AND N1.OWNER = N2.OWNER
```

関連概念:

- 260 ページの『Extended Search のニックネーム』
- 261 ページの『Extended Search の垂直表』

関連タスク:

- 268 ページの『Extended Search データ・ソースのニックネームの登録』
- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 274 ページの『Extended Search ラッパー - 汎用照会言語』
- 604 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 601 ページの『CREATE NICKNAME ステートメント構文 - Extended Search ラッパー』
- 270 ページの『Extended Search ラッパー - 照会のガイドライン』

Extended Search ラッパー - 汎用照会言語

Extended Search ラッパーによって Extended Search サーバーに渡す照会には、汎用照会言語 (GQL) である Extended Search の照会言語による検索式を含めることができます。

たとえば、従業員情報の表を含むリレーショナル・データベースにある JO で始まる名前のすべての従業員を検索するとします。GQL で次の照会を発行することができます。

```
(LIKE "EMPLOYEE_NAME" "JO")
```

SQL で同様の照会を次のように発行することができます。

```
SELECT * FROM EMP.TABLE WHERE EMPLOYEE_NAME LIKE JO%
```

SQL のように、ラッパーはインフィックス表記、つまりフィールド名と比較値との間に演算子を必要とする構文をサポートします。ネイティブ Extended Search GQL 文法は、接頭表記法、つまり評価しようとするフィールドと値の前に演算子を置く必要のある構文を使用します。

TITLE フィールドにある IBM という語を含む文書を検索する次の照会式を比較してください。

インフィックス GQL

```
("TITLE" IN "IBM")
```

接頭表記の GQL

```
(IN "TITLE" "IBM")
```

照会を Extended Search ラッパーでサブミットする場合、API は Extended Search による処理を行うためにインフィックス SQL ステートメントを接頭表記の GQL に変換します。

次の構文記述は、照会で使用できる Extended Search 文法の Backus-Naur Form 仕様を示します。

```

expr:          pattern_expr
              | bool_expr
              | field_expr
              | prox_expr

pattern_expr:  STRING
              | token_expr

token_expr:    ( TOKEN [[:CASE] [[:STEM] [[:EXACT] [[:WEIGHT "x"]
                  [[:WILD] [[:FUZZY] STRING )

bool_expr:     (expr_list bool_operator [[:WEIGHT "x"] expr )

bool_text_expr: (text_expr_list bool_operator [[:WEIGHT "x"] text_expr )

text_expr:     pattern_expr
              | bool_text_expr
              | prox_expr

text_expr_list: text_expr
               | text_expr_list text_expr

expr_list:     expr
              | expr_list expr

field_expr:    ( field_name operator_1 [[:WEIGHT "x"] text_expr )
              | ( field_name operator_2 [[:WEIGHT "x"] value )
              | ( field_name operator_3 [[:WEIGHT "x"] value_1 AND value_2 )
              | ( field_name operator_4 value )

prox_expr:     ( prox_op [[:COUNT "x"] [[:ORDER] [[:MATH "y"] [[:WEIGHT "x"]
                  expr_list expr )

prox_op:       DOCUMENT
              | PARAGRAPH
              | SENTENCE
              | WORD
              | CHARACTER

operator1:     START
              | END
              | IN
              | =

operator_2:    =
              | >
              | >=
              | <
              | <=
              | EQ
              | GT
              | GTE
              | LT
              | LTE

operator_3:    BETWEENI
              | BETWEENE
              | LIKE

```

```
bool_operator: AND
                OR
                NOT
```

SQL 文法についての詳細は、IBM Lotus Extended Search Web サイトのリソース・ページで利用できる「*Extended Search Programming*」を参照してください。

<http://www.lotus.com/products/des.nsf/wdocuments/resources>

関連タスク:

- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 272 ページの『Extended Search ラッパー - 照会の例』
- 604 ページの『CREATE FUNCTION ステートメント構文 - Extended Search ラッパー』
- 270 ページの『Extended Search ラッパー - 照会のガイドライン』

Extended Search ラッパーのメッセージ

このトピックでは、Extended Search ラッパーを使用して作業するときに出されるメッセージを説明します。

表 67. *Extended Search* ラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由: INTERNAL Extended Search WRAPPER ERROR - RC: xxx.)	理由コード (901 から 999 の番号) を記録し、IBM ソフトウェア・サポートに連絡を取ります。
SQL0973N	ステートメントの処理に使用できる十分なストレージが、アプリケーション・ヒープにありません。	Extended Search ラッパーは、アプリケーション・ヒープにメモリーを割り振ることができませんでした。問題を解決するには、アプリケーション・ヒープ・サイズを増やしてステートメントを再び試行してください。例: <pre>db2 update db cfg for db-name using heap-name heap-size</pre> <p>この値を増やしてもエラーが続く場合には、IBM ソフトウェア・サポートに連絡を取ってください。</p>

表 67. Extended Search ラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "Extended Search wrapper" から予期しないエラー・コード "<error_code>" を受け取りました。関連するテキストおよびトークンは「<tokens>」です。	検索要求の処理中に、リモート Extended Search サーバーはエラーを戻しました。さらに、エラーはリモート・サーバーでのエラーの原因を示すトークンを戻しました。Extended Search サーバーに対するトレースが使用可能になっている場合、トレース・ログ・ファイルで診断ヘルプを検討します。
SQL1823N	サーバー "<server_name>" からデータ・タイプ "<data_type>" のためのデータ・タイプ・マッピングがありません。	CREATE NICKNAME ステートメントまたは ALTER NICKNAME ステートメントにある列が、Extended Search システムがサポートしていないデータ・タイプを使用しています。このエラーは、照会処理中にも生じることがあります。この問題が照会処理中に生じる場合、解決するには、ニックネーム表をドロップして新しいニックネームを作成します。
SQL1825N	この SQL ステートメントをフェデレーテッド環境で取り扱うことはできません。	現行の SQL ステートメントは Extended Search ラッパーによって処理できません。この問題を解決するには、Extended Search ラッパーの資料を参照し、必要に応じて SQL ステートメントを変更し、要求をもう一度サブミットします。
SQL1833N	ポート "<port_number>" に、リモート拡張検索サーバー "<host_name>" との接続を確立できなかったか、またはこのサーバーが終了しました。	Extended Search ラッパーは指定されたポートでリモート Extended Search サーバーに接続しようとしたのですが、接続が確立できなかったか、リモート・サーバーによって終了されました。リモート Extended Search サーバーのホスト名とポート番号を検査し、Extended Search サーバーが稼働していることを確認し、もう一度試行してください。
SQL1834N	ユーザー定義列 "<column_name>" は、ラッパー "<wrapper_name>" の固定列と同一ですが、異なるデータ・タイプを使用しています。	CREATE NICKNAME ステートメントまたは ALTER NICKNAME ステートメントが、指定された Extended Search ラッパーの固定列と同じ名前を持つ異なるデータ・タイプを使用するユーザー定義列を含んでいます。CREATE NICKNAME ステートメントの列定義で固定列を指定する必要はありません。指定する場合は、固定列名、データ・タイプ、およびデータ・タイプの長さが固定列定義に一致することを確認してください。固定列名またはデータ・タイプは変更できません。

表 67. Extended Search ラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1835N	タイプ "<object_type>" の拡張検索オブジェクト "<object_name>" をリモート拡張検索サーバー "<host_name>" で見つけることができませんでした。	指定された Extended Search オブジェクトは、指定されたりリモート Extended Search サーバー上で検出できませんでした。オブジェクト名がこの Extended Search サーバー上で定義されていて、指定されたオブジェクト・タイプであることを確認してください。このオブジェクトのスペルが正しいことも確認してください。
SQL1836N	ユーザー定義列 "<column_name>" およびリモート拡張検索サーバー "<host_name>" のファイル名との間に列のマッピングが存在しません。	DATASOURCE または CATEGORY オプションに含まれているいずれのデータ・ソースにも、指定されたユーザー定義列の名前に一致するファイル名は含まれていません。列名が、DATASOURCE オプションのデータ・ソースの少なくとも 1 つの中にあるフィールドか、それとも CATEGORY オプションのカテゴリに属するデータ・ソースの少なくとも 1 つの中にあるフィールドであることを確認し、もう一度ステートメントをサブミットします。
SQL1837N	タイプ "<object_type>" のラッパー "<wrapper_name>" での必須オプション "<option_name>" はドロップできません。	必要なオプションをドロップできません。DROP の代わりに SET を使用するように ALTER ステートメントを変更してください。検索ステートメントを訂正し、もう一度要求をサブミットします。有効な SQL 検索ステートメントの作成について詳しくは、「DB2 UDB SQL リファレンス」を参照してください。検索ステートメントに ES_SEARCH 関数が組み込まれている場合、Extended Search 汎用照会言語 (GQL) の使用について、Extended Search ラッパーの資料を参照してください。
SQL1838N	検索ステートメント "<option_name>" が、無効な拡張検索照会です。	Extended Search ラッパーは指定された検索ステートメントを処理しようとしたのですが、そのステートメントが適切な照会構文を使用していないため、照会は失敗しました。有効な SQL 検索ステートメントの作成について詳しくは、「DB2 UDB SQL リファレンス」を参照してください。検索ステートメントに ES_SEARCH 関数が組み込まれている場合、Extended Search 汎用照会言語 (GQL) の使用について、Extended Search ラッパーの資料を参照してください。
SQL1839N	1 つ以上の検索パラメーターが無効です。	Extended Search ラッパーは指定された検索パラメーターを使用しようとしたのですが、それらのパラメーターは Extended Search に対して無効です。Extended Search ラッパーの資料を参照し、無効パラメーターを訂正し、要求をもう一度サブミットしてください。

表 67. *Extended Search* ラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1881N	"<option_name>" は "<object_name>" で有効な "<option_type>" ではありません。	指定されたオプションは、指定されたオブジェクト (ラッパー、サーバー、ニックネーム、列、またはユーザー・マッピング) に対して無効です。 <i>Extended Search</i> ラッパーの資料を参照し、無効パラメーターを除去するか変更し、要求をもう一度サブミットしてください。
SQL1882N	"<option_type>" オプション "<option_name>" が "<object_name>" について "<option_value>" に設定できません。	指定されたオプションの値は、指定されたオブジェクト (ラッパー、サーバー、ニックネーム、列、またはユーザー・マッピング) に対して無効です。 <i>Extended Search</i> ラッパーの資料を参照し、無効なオプションの値を変更し、ステートメントをもう一度サブミットしてください。
SQL1883N	"<option_name>" は "<object_name>" で必須の "<option_type>" オプションです。	<i>Extended Search</i> ラッパーに必要なオプションが、指定されたオブジェクト (ラッパー、サーバー、ニックネーム、またはユーザー・マッピング) の作成、変更、または初期化を行うステートメントに存在しませんでした。 <i>Extended Search</i> ラッパーの資料を参照し、必要なオプションの値を追加し、ステートメントをもう一度サブミットしてください。

メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。「*Extended Search Administration*」で *Extended Search* 製品メッセージを参照することもできます。不適切な GQL 照会構文についてのエラーを受け取る場合、「*Extended Search Programming*」を参照してください。*Extended Search* の資料は、IBM Lotus *Extended Search* Web サイトのリソース・ページで利用できます。

<http://www.lotus.com/products/des.nsf/wdocuments/resources>

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』
- 「メッセージ・リファレンス 第 2 巻」の『SQL0900 - SQL0999』
- 「メッセージ・リファレンス 第 2 巻」の『SQL1800 - SQL1899』

第 14 章 HMMER データ・ソースへのアクセスの構成

この章では、HMMER データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。HMMER データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- HMMER の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- HMMER ラッパーに関連するエラー・メッセージをリストします。

HMMER とは?

HMMER は、統計モデルまたはプロファイルとして Hidden Markov Model (HMM) を使用する遺伝子列データベースの検索に使用できるアプリケーション・パッケージです。HMMER アプリケーション・パッケージは、<http://hmmer.wustl.edu/> から無料でダウンロードすることができます。別々の HMMER サーバーまたはフェデレーテッド・サーバー上に、HMMER アプリケーション・パッケージをインストールできます。

HMM は、遺伝子列ファミリーの主要な構造に関するコンセンサスの統計モデルです。HMM は、確率モデルに基づいています。信頼の置ける配置がまだ解明されていない場合、配置が確認されていない遺伝子列からパターンを認識するよう、HMM に学習させることができます。プロファイルを丹念に構成していくことに比べ、学習させて優れた成果をもたらすようになった HMM を使用することのほうが、求められるスキルが低くて済み、手作業の必要性も小さくなります。学習済み HMM を使用すれば、何百というプロファイル HMM のライブラリーにアクセスして、それらを幅広くゲノム全体または Expressed Sequence Tag (EST) 分析に適用できます。

PFAM (Protein Families Database of Alignments and HMMs) は、たんぱく質ドメイン・モデルのデータベースです。HMMER アプリケーション・パッケージは、PFAM データベースの構成および使用と密接なつながりがあります。

HMMER アプリケーション・パッケージには 9 つのプログラムが含まれますが、DB2[®] Information Integrator がサポートするのは、そのプログラムのうち 2 つ (hmmfam プログラムおよび hmmsearch プログラム) だけです。

表 68. HMMER ラッパーがサポートする HMMER プログラム

HMMER プログラム	説明
hmmpfam	特定の遺伝子配列を使用して HMM データベースを検索し、そのテスト遺伝子配列が属するファミリーを判別します。各モデルが、指定された列およびモデル・データベースと一致する度合いを計算します。一致の度合いは、統計の有意性で表されます。
hmmsearch	特定の HMM プロファイルを使用して、よく似たシーケンスの一致がないかどうかシーケンス・データベースを検索します。

ユーザーまたはアプリケーションは、HMMER 固有の述部がある SQL 照会ステートメントをフェデレーテッド・サーバーに発行します。これらのステートメント内の述部は、hmmpfam または hmmsearch プログラム内のコマンド行オプションにマップされます。

HMMER ラッパーは、HMMER アプリケーション・パッケージが解釈できる形式に照会ステートメントを変換した後、その照会を実行するために hmmpfam プログラムを始動します。

特別なデーモン・プログラムが HMMER アプリケーション・パッケージがインストールされているサーバーで実行されます。このデーモンは、フェデレーテッド・サーバーからの照会要求を受信し、それを HMMER アプリケーション・パッケージへ送信します。HMMER アプリケーション・パッケージは、PFAM などのプロファイル・データベース上で照会を実行します。

図 25 に、フェデレーテッド・システムでの HMMER の動作を示します。

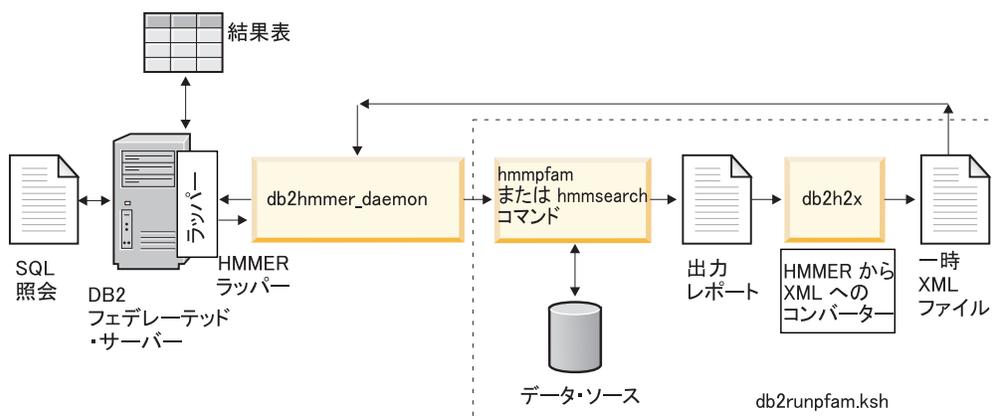


図 25. HMMER ラッパーがどのように機能するか

デーモンは、結果を HMMER ラッパーに戻します。ラッパーはデータをリレーショナル表に変換し、この表をユーザーまたはアプリケーションへ戻します。

次の例は、HMMER プログラムによって構成され、リレーショナル表として表示されるプロファイル・データベースから情報を抽出する方法を示すものです。

HMMER User's Guide <http://hmmmer.wustl.edu/> には、プロファイル・データベースの作成例と HMMER のチュートリアルが載せられています。

図 26 は、7LES_DROME 遺伝子列を使用した照会例です。遺伝子列は、照会の WHERE 文節に指定します。

```

SELECT Model, ModelScore, DomainNumber, DomainScore
FROM myhmms
WHERE HmmQSeq = 'MTMFQQNVHDHQSDEQDKQAKGAAPTAKRLNISFNVKIAVNVNTKMTTTH
INQQAPGTSSSSNSQNASPSKIVVRQQSSSFDLRQQLARLGRQLASGQDGHGGISTILINLLLL
ILLSICCDVCRSHNYTVHQSPVSKDQMRLLRPKLDSDVVEKVAIWHKHAAPPSIVEGIAISS
RPQSTMAHHPDDRDRDRDPEEQHGVDERMVLERVRTDCVQRCIVEEDLFLDFEFGIQCEKADNGEK
CYKTRCTKGCQWYRALKELESCQEACLSLQFYPYDMPICIGACEMAQRDYWHLQRLAISHLVERTQ
PQLERAPRADGQSTPLTIRWAMHFPEHYLASRPFNIQYQFVDHHEEELDLQEQDQASGETGSSAW
FNLADYDCDEYYMCEILEALIPYTYRFRFELPFGENRDEVLYSPATPAYQTPPEGAPISAPVIEH
LMGLDDSHLAVHWHGPRFTNGPIEGYRLRLSSSEGNATSEQLVPAGRGSYIFSQQLAGTNYTLALS
MINKQEGEPVAKGFVQTHSARNEKPAKDLTESVLLVGRRAVMQWSPLEPAGENSMIYQSQEELADIA
WSKREQLLWLLNVHVELRSLKFESGMVSPAQLKLDLGNISSGRWVPRRLSFDWLHHRLYFAMES
PERNQSSFQIISTDLLGESAQKVGESFDLPVEQLEVDALNGWI FWRNEESLWRQDLHGRMIHRLLR
IRQPGWFLVQPQHFIIHMLPQEGKFLIISYDGGFKHPLPLPPSNGAGNGPASHWQSFALLGRS
LLLPSGQLLVEQQGAASPSASWPLKNLPDCWAVILLVPESQPLTSAGGKPHSLKALLGAQAAK
ISWKEPERNPYQSADAARSWSYELEVLVDASQSAFIRNIRGPIFGLQRLQPDNLYQLRVRAINVD
GEPGEWTEPLAARTWPLGPHRLRWASRQGSVIHTNELGEGLEVQQEQLELPGPMTMVNESVGYVY
TGDGLLHCINLVHSQWGPCISEPLQHVGSVTYDWRGGRVYWDLARNCVVRMDPWSGSRELLPVFE
ANFLALDPRQGHLYATSSQLSRHGSTPDEAVTYRNVNGLEGSIAFVLDTQQDLFWLVKGGSGAL
RLYRAPLTAGGDSLQMIQKIGVFQAVPDSLQLLRPLGALLWLESGRRARLVRLAAPLDVMEPLPT
PDQASPASALQLLDPQLPPRDEGVIPMTVLPDSVRLDDGHWDHFVRWQVSTSGGNHSVSYRLLL
EFGQRLQTLDLSTPFARLTQLPQAQLKISITPRTAWRSGDTTRVQLTTPPVAPSQPRRLRVFE
RLATALQEANVSAVLRWDAPEQGEAPMQALEYHISCVWGSSELHEELRLNQSALAEARVEHLQPDQT
YHFQVEARVAATGAAAGAASHALHVAPEVQAVPRVLYANAEIFIGELDLDRNRRLVHTASPEHL
VGIIEGEQRLWVNEHVELLTHVPGSAPAKLARMRAEVLALAVDWIQRIVYWAELDATAPQAAIYR
LDLCNFEGKILLQGERVWSTPRGRLLKDLVALPQAQSLIWLVEYEQGSPRNGSLRGRNLDGSELEWA
TVQPLIRLHAGSLEPGSETLNLVDNQGKLCVYDVARQLCTASALRAQLNLLGEDSIAGQLAQDSGY
LYAVKNWSIRAYGRRRQQLLEYTVELEPEEVRLQAHNYQAYPPKNCLLPSGGSLKATDCEEQR
CLLNLPMITASEDCPLIPGVRYQLNLTARGPGSEEHGVEPLGQWLLGAGESLNLTDLLPFTR
YRVSGILSSFYQKALPTLVLAPLELLTASATPSPRRNFVSRVLSPRELEVSWLPPEQLRSESVY
YTLHWQQLDGENVQDRREWEAHERRLETAGHRLTGKPGSGSLWVQAHATPTKSNSSERLHVR
SFAELPELQLELGPYSLSLTWAGTPDPLGSLQLECRSSAEQLRRNVAGNHTKMVVEPLQPRTRYQ
CRLLLGYAATPGAPLYHGTAEVYETLGDAPSQPGKPLEHIAEEVFRVWTAARGNGAPIALYNLE
ALQARSDIRRRRRRRRRNSGGSLEQLPWAEPEVVEDQWDFCNTTELSICIVKSLHSSRLLLFRVR
ARSLEHGWPYSEESERVAEPFVSPEKRGSLVLAIIAPAAIVSSCVLALVLRKVQKRRRLRAKLL
QQSRPSIWSNLSTLQTTQQLMAVRNRAFSTTSDADIALLPQINWSQLKLLRFLGSGAFGEVYEGQ
LKTEDSEEPQRVAIKSLRKGASEFAELLQEAQLMSNFKHENIVRLVGCIFDTESISLIMEHMEAGD
LLSYLRAARATSTQEPQPTAGLSSELLAMCIDVANGCSYLEDMHFVHRDLACRNCLVTESTGSTD
RRRTVKIGDFGLARDIYKSDYRKEGEGLLPVRWMSPELVLDGLFTTQSDVWAFGLCWEILTGLQ
QPYAARNNFVLAHVKEGGRLLQPPMCTEKLYSLLLLCWRTDPWERPSFRRCYNTLHAISTDLRRT
QMASATADTVVSCSRPEFKVRFQDQPLEEHREHNERPEDENLTLREVPLKDKQLYANEGVSRL'
```

図 26. 7LES_DROME データに対する照会の例

HMMER ラッパーは、照会からの結果を、表 69 に示すリレーショナル表に変換します。

表 69. HMMER 結果は、リレーショナル表に変換されます。

Model	ModelScore	DomainNumber	DomainScore
pkinase	+3.04100000000000E+002	1	+3.04100000000000E+002
fn3	+1.76300000000000E+002	1	+4.90000000000000E+001
fn3	+1.76300000000000E+002	2	+1.36000000000000E+001
fn3	+1.76300000000000E+002	3	+1.62000000000000E+001
fn3	+1.76300000000000E+002	4	+6.35000000000000E+001
fn3	+1.76300000000000E+002	5	+1.46000000000000E+001

表 69. HMMER 結果は、リレーショナル表に変換されます。(続き)

Model	ModelScore	DomainNumber	DomainScore
fn3	+1.76300000000000E+002	6	+1.94000000000000E+001
rrm	-4.45000000000000E+001	1	-4.45000000000000E+001

これでデータはリレーショナル形式になり、その他のデータ・ソースからのデータと結合できるようになりました。

関連タスク:

- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』

フェデレーテッド・サーバーへの HMMER の追加

フェデレーテッド・サーバーへの HMMER データ・ソースの追加

フェデレーテッド・サーバーが HMMER データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供する必要があります。

DB2 コントロール・センター、または DB2 コマンド行を使用して、HMMER データ・ソースにアクセスできるようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

HMMER データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. HMMER プログラム実行可能ファイルの正しいバージョンがインストールされていることを検証します。
2. HMMER デーモンを構成します。
3. HMMER デーモンを開始します。
4. ラッパーを登録します。
5. サーバー定義を登録します。
6. ニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator のインストール - 概要』
- 281 ページの『HMMER とは?』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 285 ページの『HMMER プログラムの実行可能ファイルのバージョンの検査』
- 286 ページの『HMMER デーモンの構成』
- 293 ページの『HMMER ラッパーの登録』
- 294 ページの『HMMER データ・ソースのサーバー定義の登録』
- 296 ページの『HMMER データ・ソースのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』
- 302 ページの『HMMER データ・ソース - 完全指定の例』

HMMER プログラムの実行可能ファイルのバージョンの検査

必要なバージョンの HMMER 実行可能プログラムがインストールされているかどうかの検査は、HMMER データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成します。

サポートされるバージョンの hmmpfam および hmmsearch 実行可能ファイルを、HMMER アプリケーション・プログラムがインストールされているサーバー上にインストールする必要があります。HMMER アプリケーション・プログラムは、フェデレーテッド・サーバーまたは別個の HMMER サーバー上にインストールできます。

手順:

実行可能ファイルのバージョンをチェックするには、次のようにします。

1. バージョン番号を戻す以下のコマンドを発行します。
 - hmmpfam プログラムの場合、コマンドは以下のとおりです。

```
hmmpfam -h
```
 - hmmsearch プログラムの場合、コマンドは以下のとおりです。

```
hmmsearch -h
```
2. 出力ファイル内で、実行可能ファイルのバージョンをチェックします。HMMER バージョンは 2.2g (またはそれ以降) でなければなりません。
3. 正しいバージョンでない場合は、<http://hmmer.wustl.edu/> からファイルをダウンロードしてください。

この一連の作業における次のタスクは、『HMMER デーモンの構成』です。

関連タスク:

- 286 ページの『HMMER デーモンの構成』

HMMER デーモンの構成

HMMER デーモンの構成は、HMMER をフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

HMMER ラッパーは HMMER デーモンを必要とします。HMMER デーモンは、DB2 Universal Database フェデレーテッド・システムから TCP/IP を使用してアクセス可能なサーバー上で実行されなければなりません。これは、フェデレーテッド・サーバーが実行されるのと同じサーバーか、別個の HMMER サーバーにすることができます。

デーモンは、ラッパーと DB2 Universal Database とは別に実行されます。このデーモンは、ラッパーからの HMMER ジョブ要求を listen します。

前提条件:

HMMER デーモンには、以下の権限が必要です。

- HMMER 検索を実行するための、hmpfam および hmmsearch 実行可能ファイルの実行アクセス権限。
- 一時ファイルに書き込むことができるディレクトリーに対する書き込みアクセス権限。
- HMMER 検索を実行できる、少なくとも 1 つのプロファイル・データベースへの読み取りアクセス権限。

制約事項:

HMMER デーモンは、実行可能ファイルまたはデータベース・パスにスペースが含まれる場合、正しく動作しない場合があります。たとえば、Windows サーバー上の C:\Program Files に HMMER 実行可能ファイルをインストールしないでください。

手順:

HMMER デーモンを構成するには、以下のように行います。

1. HMMER デーモン実行可能ファイルが正しいサーバー上にあることを確認してください。DB2 Information Integrator のインストール時に、デーモン実行可能ファイルはフェデレーテッド・サーバー上の以下のディレクトリーにインストールされます。

UNIX の場合

デーモン実行可能ファイルは \$DB2PATH/bin ディレクトリーにインストールされます。

Windows の場合

デーモン実行可能ファイルは %DB2PATH%\bin ディレクトリーにインストールされます。

別個の HMMER サーバーを使用する場合、デーモン実行可能ファイルを、フェデレーテッド・サーバー上のディレクトリーから HMMER サーバー上のディレクトリーにコピーする必要があります。デーモン実行可能ファイルは、ディレクトリー・パスの名前にスペースが含まれない HMMER サーバー上の任意のディレクトリーで実行することができます。

2. 構成ファイルおよび他の必要なファイルが、HMMER がインストールされているサーバー上にあることを確認してください。必要なファイルの一部は、フェデレーテッド・サーバー上のディレクトリーに DB2 Information Integrator とともにインストールされています。他の必要なファイルを指定する必要があります。

UNIX の場合

HMMER がインストールされているサーバー上になければならないファイルは、以下のとおりです。

- デーモン実行可能ファイル、%DB2PATH%/bin/db2hmmmer_daemon
- HMMER デーモン構成ファイル、
%DB2PATH%/samples/lifesci/HMMER_DAEMON.config
- 変換ユーティリティー、%DB2PATH%/bin/db2h2x
- シェル・スクリプト、%DB2PATH%/bin/db2runpfam.ksh
- HMMER 実行可能ファイル (IBM 提供ではない)、hmmpfam および
hmmsearch
- HMMER データベース・ファイル (IBM 提供ではない)

%DB2PATH% は、DB2 Information Integrator がインストールされているパスです。

Windows の場合

HMMER がインストールされているサーバー上になければならないファイルは、以下のとおりです。

- デーモン実行可能ファイル、%DB2PATH%\%bin%\db2hmmmer_daemon.exe および
%DB2PATH%\%bin%\db2hmmmer_daemon_svc.exe
- HMMER デーモン構成ファイル、
%DB2PATH%\%samples%\lifesci%\HMMER_DAEMON.config
- 変換ユーティリティー、%DB2PATH%\%bin%\db2h2x.exe
- HMMER 実行可能ファイル (IBM 提供ではない)、hmmpfam.exe およ
び hmmsearch.exe
- HMMER データベース・ファイル (IBM 提供ではない)

%DB2PATH% は、DB2 Information Integrator がインストールされているパスです。

デフォルトでは、デーモンは、デーモンが開始される作業ディレクトリー内で構成ファイルを検索します。構成ファイルは、別のロケーションにコピーすることができます。HMMER サーバーを使用する場合、デーモン構成ファイルを、フェデレーテッド・サーバー上のディレクトリーから HMMER サーバー上のディレクトリーにコピーする必要があります。デーモン構成ファイルは、デーモンがアクセスできる HMMER サーバー上の任意のディレクトリーにコピーすることができます。

3. UNIX の場合、HMMER デーモン実行可能ファイル、変換ユーティリティー、およびシェル・スクリプトが実行可能であることを確認してください。ファイルを実行可能にするには、以下のコマンドを実行してください。

```
chmod a+x db2hmmmer_daemon db2h2x db2runpfam.ksh
```

4. デモン構成ファイルを編集して、データ・ソースを扱うことができるようにします。構成ファイルを名前変更することもできます。
 - 構成ファイルの 1 行目は、等号でなければなりません。等号がないと、デモンは開始しません。 `DAEMON_PORT` が指定されなかったことを示すエラー・メッセージが出されます。
 - 構成ファイルの最終行は、改行で終了する必要があります。 `DB2 Information Integrator` に付属のサンプル構成ファイルは、改行で終了しています。ファイルを編集する場合、ファイルの最終行は改行で終了する必要があります。最終行が改行で終了していない場合、最終行にリストされたデータ・ソースを使用して `HMMER` 照会を初めて実行しようとする際に、エラー・メッセージが出されます。
 - 構成ファイルに以下のオプションを指定します。パスを必要とするオプションについては、相対パスを指定できます。相対パスは、デモン・プロセスが開始されたディレクトリーに相対しています。

DAEMON_PORT

これは、ラッパーによってサブミットされた `HMMER` ジョブ要求をデモンが `listen` するネットワーク・ポートです。

MAX_PENDING_REQUESTS

これは、任意の一時点でデモンでブロックできる `HMMER` ジョブ要求の最大数です。この数は、並行して実行される `HMMER` ジョブ数を表すものではありません。一時点でブロックできるジョブ要求数のみです。これを 5 より大きい数に設定することをお勧めします。

`HMMER` デモンでは、並行して実行できる `HMMER` ジョブ数の制限はありません。

DAEMON_LOGFILE_DIR

これは、デモンがそのログ・ファイルを作成するディレクトリーです。このファイルには、`HMMER` デモンが生成する有用な状況情報とエラー情報が含まれます。

Q_SEQ_DIR_PATH

これは、デモンによって一時照会シーケンス・データ・ファイルが作成されるディレクトリーです。この一時ファイルは、`HMMER` ジョブが完了するとクリーンアップされます。

HMMER_OUT_DIR_PATH

これは、デモンが `HMMER` 出力データを保管するための一時ファイルを作成するディレクトリーです。データはこのファイルから読み取られて、ネットワーク接続を介してラッパーに戻されます。データがラッパーに渡されると、デモンは一時ファイルをクリーンアップします。

RUNPFAM_PATH

これは、`DB2 Information Integrator` に付属している `db2runpfam.ksh` シェル・スクリプトの完全修飾名です。 `Windows` で指定される場合、このオプションは無視されます。

HMMERPFAM_PATH

これは、デモンを実行しているコンピューター上の `HMMER` 実行可

能ファイルの完全修飾名です。UNIX の場合、このファイルの名前は hmmpfam です。Windows の場合、このファイルの名前は hmmpfam.exe です。

HMMSEARCH_PATH

これは、デーモンを実行しているコンピューター上の HMMER 実行可能ファイルの完全修飾名です。UNIX の場合、このファイルの名前は hmmsearch です。Windows の場合、このファイルの名前は hmmsearch.exe です。

H2X_PATH

これは、デーモンに付属している (HMMER から XML への) 変換プログラムの完全修飾名です。UNIX の場合、このプログラムの名前は db2h2x です。Windows の場合、このプログラムの名前は db2h2x.exe です。

データベース指定項目

プロファイル・データベースまたはシーケンス・ファイルのロケーションを指定します。構成ファイルに指定したデータベース *data_source_name* をメモしてください。デーモンが正しく機能するためには、データ・ソースのニックネームを作成するときにデータベース *data_source_name* を指定する必要があります。この名前は大文字小文字が区別されます。データベース *data_source_name* は、以下で指定されま

- CREATE NICKNAME ステートメントの DATASOURCE オプション (hmmpfam の場合)
- CREATE NICKNAME ステートメントの MODEL 述部 (hmmsearch の場合)

構成ファイルには、少なくとも 1 つのデータベース指定項目が、次のフォームで含まれている必要があります。

data_source_name=fully_qualified_name_of_profile_or_sequence_database

UNIX の場合

たとえば、MYHMMS プロファイル・データベースを指定するには、次の行をデーモン構成ファイルに追加します。

```
myhmms=/home/user_ID/myhmms
```

Windows の場合

たとえば、MYHMMS プロファイル・データベースを指定するには、次の行をデーモン構成ファイルに追加します。

```
myhmms=c:\hmmertutorial\myhmms
```

この一連の作業における次のタスクは、『HMMER デーモンの開始』です。

関連タスク:

- 290 ページの『HMMER デーモンの開始』

関連資料:

- 290 ページの『HMMER デーモン構成ファイル - 例』

HMMER デーモン構成ファイル - 例

以下の例は、PFAM および SEARCH 用のサンプル構成ファイルの内容を示しています。

例 - HMMER_DAEMON.config ファイル (UNIX の場合):

この例は、UNIX の場合の必須オプションとプロファイル・データベース仕様を示しています。

```
=
DAEMON_PORT=4098
MAX_PENDING_REQUESTS=10
DAEMON_LOGFILE_DIR=./
Q_SEQ_DIR_PATH=./
HMMER_OUT_DIR_PATH=./
RUNPFAM_PATH=./db2runpfam.ksh
HMMPFAM_PATH=/home/user_id/hmmer/bin/hmmpfam
HMMSEARCH_PATH=/home/user_id/hmmer/bin/hmmsearch
H2X_PATH=/home/user_id/sql1lib/bin/db2h2x
myhmms=/home/user_id/hmmer/tutorial/myhmms
globin=/home/user_id/hmmer/tutorial/globin.hmm
pfam1s=/home/user_id/hmmer/pfam/Pfam_1s
```

例 - HMMER_DAEMON.config ファイル (Windows の場合):

この例は、Windows の場合の必須オプションとプロファイル・データベース仕様を示しています。

```
=
DAEMON_PORT=4098
MAX_PENDING_REQUESTS=10
DAEMON_LOGFILE_DIR=.%
Q_SEQ_DIR_PATH=.%
HMMER_OUT_DIR_PATH=.%
HMMPFAM_PATH=c:%hmmer%bin%hmmpfam.exe
HMMSEARCH_PATH=c:%hmmer%bin%hmmsearch.exe
H2X_PATH=.%db2h2x.exe
myhmms=c:%hmmer%tutorial%myhmms
globin=c:%hmmer%tutorial%globin.hmm
pfamseq=c:%hmmer%pfam%pfamseq
```

関連タスク:

- 286 ページの『HMMER デーモンの構成』

HMMER デーモンの開始

HMMER デーモンの開始は、フェデレーテッド・サーバーに HMMER データ・ソースを追加するという、より大きな作業の一部を成しています。HMMER デーモンを開始するまで、HMMER データ・ソースにはアクセスできません。

前提条件:

HMMER デーモンを開始するには、構成ファイルの中の DAEMON_LOGFILE_DIR、HMMER_OUT_DIR_PATH、および Q_SEQ_DIR_PATH 項目の下にリストされているすべてのパスに対する書き込みアクセス権限をもっている必要があります。

手順:

UNIX サーバー上で HMMER デーモンを開始するには、以下のようにします。

1. デーモン実行可能ファイルが存在するディレクトリーを開きます。
2. 以下の `db2hmmmer_daemon` コマンドを発行します。

- デーモン構成ファイルの名前を変更しておらず、構成ファイルがデーモン実行可能ファイルと同じディレクトリーにある場合、コマンド行に以下のコマンドを入力します。

```
db2hmmmer_daemon
```

- デーモン構成ファイルの名前を変更してしまっているか、構成ファイルがデーモン実行可能ファイルと同じディレクトリーにない場合、ラッパー・デーモン・コマンドで `-c` オプションを使用して、デーモン実行可能ファイルが新しい名前または新しい場所を指すようにする必要があります。

たとえば、以下のコマンドは、ラッパー・デーモンがデーモン構成情報をサブディレクトリー `cfg` の中の `HMMER_D.config` というファイルの中で探すようにさせます。

```
db2hmmmer_daemon -c cfg/HMMER_D.config
```

実行可能ファイルにより、HMMER デーモンが実行される新しいプロセスが開始されます。

UNIX サーバー上でデーモンを停止するには、以下のようにします。

1. 以下の UNIX コマンドを使用して、`db2hmmmer_daemon` のプロセス ID をリストします。

```
ps -ef | grep db2hmmmer
```

2. プロセス ID を使用してデーモンを停止します。以下のコマンドを使用します。

```
kill nnnn
```

ここで、`nnnn` は `db2hmmmer_daemon` のプロセス ID です。

Windows サーバー上で HMMER デーモンを開始するには、以下のようにします。

1. デーモン実行可能ファイルが存在するディレクトリーを開きます。
2. 必要とするパラメーターが含まれる `db2hmmmer_daemon` コマンドを発行します。たとえば、デバッグをオンにした状態でデーモン・サービスをインストールし、デーモンを開始するには、以下のコマンドを発行します。

```
db2hmmmer_daemon -a install -d 2
```

```
db2hmmmer_daemon -a start
```

デーモンを停止するには、以下の Windows コマンドを使用します。

```
db2hmmmer_daemon -a stop
```

この一連の作業における次のタスクは、『HMMER ラッパーの登録』です。

関連タスク:

- 286 ページの『HMMER デーモンの構成』
- 293 ページの『HMMER ラッパーの登録』

- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』

関連資料:

- 292 ページの『db2hmmer_daemon コマンド - 構文および例』

db2hmmer_daemon コマンド - 構文および例

db2hmmer_daemon コマンドは、UNIX および Windows サーバー上で使用できます。構文内でリストされる引き数の一部は、Windows サーバーでのみ使用できます。

db2hmmer_daemon コマンドの構文は以下のとおりです。

```
db2hmmer_daemon -a action -c config_file -d debug_level  
-u user_id -p password
```

-a action

指定したアクティビティを実行します。有効なアクションは、*status*、*install*、*start*、*stop*、および *remove* です。

この引き数は、Windows サーバーでのみ指定できます。

-c config_file

デフォルトの構成ファイルではなく、指定した構成ファイルを使用するようデーモン・サービスに指示します。構成ファイルを指定しなかった場合、デーモンは、デーモン実行可能ファイルがインストールされているディレクトリ内で HMMER_DAEMON.config ファイルを検索します。このオプションは、*install* および *start* アクションと一緒に使用できます。

この引き数は、UNIX および Windows サーバーで指定できます。

-d debug_level

デーモン・サービスのデバッグ・レベルを、指定した値に設定します。有効な値は、1、2、あるいは 3 です。このオプションは、*install* および *start* アクションと一緒に使用できます。

この引き数は、UNIX および Windows サーバーで指定できます。

-u user_id

指定したユーザー ID で実行するサービスを設定します。このオプションは、*install* アクションと一緒に使用できます。

この引き数は、Windows サーバーでのみ指定できます。

-p password

指定したユーザー ID のパスワードを指定します。パスワードが有効かつ必要なのは、*-u* オプションを指定する場合だけです。*-u* オプションの設定時に *-p* オプションを指定しなかった場合、パスワードを求めるプロンプトがプログラムによって出されます。このオプションは、*install* アクションと一緒に使用できます。

この引き数は、Windows サーバーでのみ指定できます。

start アクションで指定されるオプションは、現在のデーモンの実行のみに影響し、*install* アクションを使用して指定される値をオーバーライドします。

例:

以下の例は、Windows 上のデーモン・アクションを示しています。これらの例は、HMMER_DAEMON.config ファイルが db2hmmmer_daemon.exe と同じディレクトリー内にあることを前提としています。

- デーモンの状況をチェックするには、次のようにします。

```
db2hmmmer_daemon -a status
```

- デバッグをオンにした状態でデーモン・サービスをインストールするには、次のようにします。

```
db2hmmmer_daemon -a install -d 2
```

- デーモンを開始するには、次のようにします。

```
db2hmmmer_daemon -a start
```

- デーモンを停止するには、次のようにします。

```
db2hmmmer_daemon -a stop
```

- デーモン・サービスを除去またはアンインストールするには、次のようにします。

```
db2hmmmer_daemon -a remove
```

関連タスク:

- 290 ページの『HMMER デーモンの開始』

HMMER ラッパーの登録

HMMER ラッパーの登録は、フェデレーテッド・サーバーに HMMER データ・ソースを追加するという、より大きな作業の一部を成しています。

HMMER データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で `hmmmer_wrapper` という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER hmmmer_wrapper LIBRARY 'libdb2lshmmmer.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『HMMER ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『HMMER ラッパーのサーバー定義の登録』です。

関連タスク:

- 294 ページの『HMMER データ・ソースのサーバー定義の登録』

関連資料:

- 294 ページの『HMMER ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

HMMER ラッパー・ライブラリー・ファイル

次の表は、HMMER ラッパーのディレクトリー・パスおよびライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2lshmmmer.a、libdb2lshmmmerF.a、および libdb2lshmmmerU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 70. HMMER ラッパー・ライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lshmmmer.a
Linux	/opt/IBM/db2/V8.1/lib	libdb2lshmmmer.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lshmmmer.so
Windows	%DB2PATH%¥bin	db2lshmmmer.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、C:¥Program Files¥IBM¥SQLLIB です。

関連タスク:

- 293 ページの『HMMER ラッパーの登録』

HMMER データ・ソースのサーバー定義の登録

HMMER データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに HMMER データ・ソースを追加するという、より大きな作業の一部を成しています。

ラッパーの登録後、対応するサーバー定義を登録する必要があります。

手順:

HMMER サーバー定義を登録するには、CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE HMMER_search_type
  VERSION version WRAPPER wrapper_name
  OPTIONS (NODE 'node_name', DAEMON_PORT 'port_number')
```

HMMER 検索を実行したい各サーバーごとに定義を登録する必要があります。

この一連の作業における次のタスクは、『HMMER データ・ソースのニックネームの登録』です。

関連タスク:

- 296 ページの『HMMER データ・ソースのニックネームの登録』

関連資料:

- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 295 ページの『CREATE SERVER ステートメント - HMMER ラッパーの例』

CREATE SERVER ステートメント - HMMER ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して HMMER ラッパー用サーバー定義を登録する方法を示す例が提供されます。

hmmpfam プログラムを使用する検索用のサーバー定義 `hmmpfam_server` を登録するには、以下のステートメントを発行します。

```
CREATE SERVER hmmpfam_server
  TYPE pfam
  VERSION 2.2
  WRAPPER hmmer_wrapper
  OPTIONS (NODE 'someserver.someschool.edu', DAEMON_PORT '4098')
```

hmmpfam_server

HMMER サーバー定義に割り当てる名前。この名前は固有でなければなりません。

TYPE *pfam*

サーバー定義がサポートする検索のタイプ。指定できるタイプは、PFAM (hmmpfam の場合) または SEARCH (hmmsearch の場合) です。

VERSION 2.2

使用している hmmpfam または hmmsearch 実行可能ファイルのバージョン。サポートされるバージョンは HMMER 2.2g (あるいはそれ以降) です。

WRAPPER *hmmer_wrapper*

CREATE WRAPPER ステートメントで指定した名前。

NODE *'someserver.someschool.edu'*

HMMER デーモン・プロセスが実行されるサーバーのホスト名または IP アドレス。

ノード名は CREATE SERVER ステートメントでオプションですが、HMMER データ・ソースの場合は必要です。

DAEMON_PORT *'4098'*

デーモンが HMMER ジョブ要求を listen するポート番号。ポート番号は、

デーモン構成ファイルの DAEMON_PORT オプションに指定されているものと同じ番号でなければなりません。デフォルトは 4098 です。

追加のサーバー・オプション:

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。

PROCESSORS

HMMER プログラムが使用するプロセッサの番号。このオプションは、**hmmpfam** および **hmmsearch** コマンドの `--cpu` オプションと同等です。
例: PROCESSORS '2'。

HMMPFAM_OPTIONS

このサーバー・オプションを使用して、述部で指定できない **hmmpfam** コマンドにオプションを渡します。例: HMMPFAM_OPTIONS '--null12 --pvm'。この例では、hmmpfam は、照会がこのサーバーに対して実行される場合、常に 2 つの追加オプション、--null12 および --pvm を指定して実行されます。HMMPFAM_OPTIONS オプションは、タイプ PFAM として指定されるサーバーでのみ有効です。

HMMSEARCH_OPTIONS

このサーバー・オプションを使用して、述部で指定できない **hmmsearch** コマンドにオプションを渡します。例: HMMSEARCH_OPTIONS '--null12 --pvm'。この例では、hmmsearch は、照会がこのサーバーに対して実行される場合、常に 2 つの追加オプション、--null12 および --pvm を指定して実行されます。HMMSEARCH_OPTIONS オプションは、タイプ SEARCH として指定されるサーバーでのみ有効です。

関連タスク:

- 294 ページの『HMMER データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』

HMMER データ・ソースのニックネームの登録

HMMER データ・ソースのニックネームの登録は、フェデレーテッド・サーバーに HMMER データ・ソースを追加するという、より大きな作業の一部を成しています。

サーバー定義を登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で HMMER データ・ソースを参照する場合に使用します。

手順:

HMMER ニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。例えば、以下のようになります。

```
CREATE NICKNAME nickname
  FOR SERVER server_name
  OPTIONS(DATASOURCE 'data_source')
```

照会する各プロファイル・データベースごとに別個のニックネームを定義する必要があります。 `data_source` 名は、HMMER サーバー上の `HMMER_DAEMON.config` ファイル内の既存の `data_source_name` と一致している必要があります。

HMMER データベースのニックネームを作成すると、プロファイル・データベースの入力および出力 固定列がフェデレーテッド・データベース・システム・カタログ内に登録されます。

一連の作業はこれで終わりです。

関連タスク:

- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』
- 286 ページの『HMMER デーモンの構成』
- 304 ページの『サンプルを使用した新規 HMMER 照会の構成』
- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』

関連資料:

- 297 ページの『HMMER ニックネームの固定列』
- 302 ページの『HMMER データ・ソース - 完全指定の例』
- 301 ページの『CREATE NICKNAME ステートメント - HMMER ラッパーの例』
- 297 ページの『HMMER ニックネームの固定列』

HMMER ニックネームの固定列

HMMER データ・ソースに対して `CREATE NICKNAME` ステートメントを発行すると、そのニックネームで固定入力列および固定出力列が自動的に作成されます。固定列に割り当てるデフォルトのデータ・タイプを変更したい場合、`CREATE NICKNAME` ステートメント内で列名およびデータ・タイプを指定できます。たとえば、`AlignmentConsensus` 列の出力を最初の 100 文字のみに制限するには、以下のステートメントを発行します。

```
CREATE NICKNAME nucleo1 (AlignmentConsensus VARCHAR(100))
FOR SERVER searchtest
OPTIONS(DATASOURCE 'nucleo1', TIMEOUT '1');
```

これらの固定列は、SQL 照会で、ニックネーム定義の一部として参照できます。固定列には入力と出力の 2 つのタイプがあります。

HMMER ニックネームの固定入力列

固定入力列は `WHERE` 文節で指定されます。入力列は、SQL 照会のパラメーター受け渡し述部として使用されます。それら固定列は、標準の HMMER スイッチを `hmmfam` または `hmmsearch` のいずれかに渡します。次に、これらのスイッチを使用して、指定されたデータ・ソースに対して HMMER が実行されます。固定入力列は、照会の `SELECT` リストの中で参照することもでき、結果表の一部として戻されます。

タイプ **PFAM** のサーバー用固定入力列:

以下の表は、WHERE 文節で使用できる固定列をリストしています。

表 71. タイプ PFAM のサーバー用固定入力列

名前	データ・ タイプ	説明	演算子	スイッチ	戻り値
HmmQSeq	varchar(32000)	検索に使用される入力遺伝子列。	=		指定した入力値と同じ。この列は必須です。
ModelEValue	double	見積もられる e 値	<	-E <i>n</i>	出力を参照してください。
ModelScore	double	ロー・スコア	>	-T <i>n</i>	出力を参照してください。
DBSize	integer	データベースに 'n' 個の遺伝子列があるとして計算された e 値	=	-Z <i>n</i>	指定した入力値と同じ。指定されない場合は、デフォルトの hmmpfam が使用されます。
CutMode	char(2)	カットオフ・モード - ga、tc、または nc (大文字小文字の区別あり)	=	--cut_ga --cut_tc --cut_nc	指定した入力値と同じ。指定されない場合は NULL になります。
DomainScore	double	ドメイン・スコア	>	--domT <i>n</i>	出力を参照してください。
DomainEValue	double	ドメイン e 値	<	--domE <i>n</i>	出力を参照してください。
ForwardAlgorithm	char	Viterbi アルゴリズムではなく、Forward アルゴリズムを使用 (値は 'Y' または 'N')	=	--forward	指定した入力値と同じ。'N' はデフォルトです。

タイプ SEARCH のサーバー用固定入力列:

以下の表は、WHERE 文節で使用できる固定列をリストしています。

表 72. タイプ SEARCH のサーバー用固定入力列

名前	データ・ タイプ	説明	演算子	オプション	戻り値
モデル	varchar (32000)	検索で使用される HMM プロファイル・ ファイルの名前。この 名前は、 HMMER_DAEMON.config ファイル内のデータベ ース仕様にリストされ るデータ・ソース名の いずれかでなければな りません。	=		指定した入 力値と同 じ。この列 は必須で す。
SequenceEValue	double	見積もられる e 値	<	-E <i>n</i>	出力を参照 してください。
SequenceScore	double	ロー・スコア	>	-T <i>n</i>	出力を参照 してください。
DBSize	integer	データベースに 'n' 個 の遺伝子列があるとし て計算された e 値	=	-Z <i>n</i>	指定した入 力値と同 じ。指定さ れない場合 は、デフォ ルトの hmmpfam が使用され ます。
CutMode	char(2)	カットオフ・モード - ga、tc、または nc (大 文字小文字の区別あ り)	=	--cut_ga --cut_tc --cut_nc	指定した入 力値と同 じ。指定さ れない場合 は NULL になりま す。
DomainScore	double	ドメイン・スコア	>	--domT <i>n</i>	出力を参照 してください。
DomainEValue	double	ドメイン e 値	<	--domE <i>n</i>	出力を参照 してください。
ForwardAlgorithm	char	Viterbi アルゴリズム ではなく、Forward ア ルゴリズムを使用 (値 は 'Y' または 'N')	=	--forward	指定した入 力値と同 じ。'N' は デフォルト です。

HMMER ニックネームの固定出力列

SELECT リスト内の固定出力列のいずれかを指定できます。(述部としての) WHERE 文節内に固定出力列を指定することもできます。

PFAM の固定出力列:

以下の表は、PFAM の出力として戻される固定列をリストしています。

表 73. PFAM の固定出力列

名前	データ・タイプ	説明
モデル	varchar(32)	モデルの名前。
ModelDescription	varchar(64)	モデルに関するテキスト記述。
ModelScore	double	ロー・スコア ("ビット・スコア")。
ModelEValue	double	見積もられる e 値。
ModelHits	integer	モデル内でのドメインのヒット数。
DomainNumber	integer	特定のドメイン (1 つのモデル内)。
SequenceFrom	integer	遺伝子列の開始点。
SequenceFromGlobal	char	配置が遺伝子列の先頭から始まる場合は 'Y'。
HmmFrom	integer	コンセンサス・モデルの開始点。
HmmFromGlobal	char	配置がコンセンサス・モデルの先頭から始まる場合は 'Y'。
HmmTo	integer	コンセンサス・モデルの終了点。
HmmToGlobal	char	配置がコンセンサス・モデルの最後で終わる場合は 'Y'。
DomainScore	double	分離ドメインのロー・スコア ("ビット・スコア")。
DomainEValue	double	分離ドメインに予測される値。
AlignmentConsensus	varchar(32000)	HMM コンセンサス。HMM によれば、コンセンサスに表示されるアミノ酸は、この位置にくる可能性が最も高いアミノ酸ですが、これは必ずしもスコアの最も高いアミノ酸であるとは限りません。
AlignmentExactMatch	varchar(32000)	HMM で最も可能性の高い剰余と一致します。
AlignmentSubSequence	varchar(32000)	遺伝子列そのものを示します。

SEARCH の固定出力列:

以下の表は、SEARCH の出力として戻される固定列をリストしています。

表 74. SEARCH の固定出力列

名前	データ・タイプ	説明
シーケンス	varchar(32)	シーケンス ID。
SequenceDescription	varchar(64)	シーケンスのテキスト記述。

表 74. SEARCH の固定出力列 (続き)

名前	データ・タイプ	説明
SequenceScore	double	ロー・スコア ("ビット・スコア").
SequenceEValue	double	見積もられる e 値。
SequenceHits	integer	シーケンス内でのドメインのヒット数。
DomainNumber	integer	特定のドメイン (1 つのシーケンス内)。
SequenceFrom	integer	遺伝子列の開始点。
SequenceFromGlobal	char	配置が遺伝子列の先頭から始まる場合は 'Y'。
HmmFrom	integer	コンセンサス・モデルの開始点。
HmmFromGlobal	char	配置がコンセンサス・モデルの先頭から始まる場合は 'Y'。
HmmTo	integer	コンセンサス・モデルの終了点。
HmmToGlobal	char	配置がコンセンサス・モデルの最後で終わる場合は 'Y'。
DomainScore	double	分離ドメインのロー・スコア ("ビット・スコア")。
DomainEValue	double	分離ドメインに予測される値。
AlignmentConsensus	varchar(32000)	HMM コンセンサス。HMM によれば、コンセンサスに表示されるアミノ酸は、この位置にくる可能性が最も高いアミノ酸ですが、これは必ずしもスコアの最も高いアミノ酸であるとは限りません。
AlignmentExactMatch	varchar(32000)	HMM で最も可能性の高い剰余と一致します。
AlignmentSubSequence	varchar(32000)	遺伝子列そのものを示します。

関連タスク:

- 296 ページの『HMMER データ・ソースのニックネームの登録』

関連資料:

- 302 ページの『HMMER データ・ソース - 完全指定の例』
- 301 ページの『CREATE NICKNAME ステートメント - HMMER ラッパーの例』

CREATE NICKNAME ステートメント - HMMER ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して HMMER データ・ソース用のニックネームを登録する方法を示す例が提供されます。

サーバー定義 hmmpfam_server を使用するニックネーム hmmpfam_nickname を登録するには、以下のステートメントを発行します。

```
CREATE NICKNAME hmmpfam_nickname
FOR SERVER hmmpfam_server
OPTIONS(DATASOURCE 'myhmms',TIMEOUT '30')
```

hmmpfam_nickname

ニックネームに割り当てる名前。この名前は固有でなければなりません。

SERVER *hmmpfam_server*

このニックネームを関連付けるサーバー定義の名前。

DATASOURCE *'myhmms'*

HMMER 検索を実行するデータベースの名前。このデータベースは、HMMER デーモン構成ファイル内にリストされている必要があります。

データ・ソースは **CREATE NICKNAME** ステートメントでオプションとなっていますが、HMMER データ・ソースの場合は必要です。

TIMEOUT *'30'*

ラッパーがデーモンからの結果を待つ最大時間 (分単位)。デフォルトは 60 分です。

関連タスク:

- 296 ページの『HMMER データ・ソースのニックネームの登録』

関連資料:

- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』

HMMER データ・ソース - 完全指定の例

このトピックでは、HMMER データ・ソースをフェデレーテッド・サーバーに追加するために発行する必要のある、すべての SQL ステートメントの例を示しています。またこの例では、SQL ステートメント内で作成されるニックネームを使用して実行される照会を示します。

HMMER データ・ソースの SQL ステートメントには、標準 HMMER オプションをプログラム実行ファイルに渡すのに使用される特殊な入力述部が含まれる必要があります。有効であるためには、HMMER ラッパーへ渡されるすべての照会は、少なくとも `HmmQSeq` 入力述部 (TYPE `PFAM` の場合) あるいは `model` 述部 (TYPE `SEARCH` の場合) を含んでいなければなりません。他のすべての述部はオプションです。

ニックネームに対して HMMER 照会を構成するには、**WHERE** 文節内の入力列および **SELECT** リスト内の出力列を指定します。

hmmpfam プログラムの場合の例:

この例では、`hmmpfam` プログラム用の AIX フェデレーテッド・サーバー上にラッパー、サーバー定義、およびニックネームが作成されます。またこの例では、検索シーケンス用のストリング・リテラルを使用した照会を実行します。

```
CREATE WRAPPER hmmer_wrapper
  LIBRARY 'libdb21shmmmer.a';

CREATE SERVER hmmpfam_server
  TYPE pfam VERSION 2.2
  WRAPPER hmmer_wrapper
  OPTIONS (NODE 'HMMERserv.MyCompany.com');
```

```

CREATE NICKNAME hmmpfam_nickname
  FOR SERVER hmmpfam_server
  OPTIONS(DATASOURCE 'myhmms', TIMEOUT '1');

-- Run the 7LES_DROME gene sequence on the hmmpfam_nickname
SELECT Model, substr(ModelDescription,1,50) as ModelDescription,
  ModelScore, ModelEValue, ModelHits, DomainNumber,
  SequenceFrom, SequenceTo, SequenceFromGlobal, SequenceToGlobal,
  HmmFrom, HmmTo, HmmFromGlobal, HmmToGlobal,
  DomainScore, DomainEValue,
  length(HmmQSeq) as "length(HmmQSeq)",
  length(AlignmentConsensus) as "length(AConsensus)",
  length(AlignmentMatch) as "length(AMatch)",
  length(AlignmentSubSeq) as "length(ASubSeq)",
  substr(HmmQSeq,1,64) as HmmQSeq,
  substr(AlignmentConsensus,1,64) as AlignmentConsensus,
  substr(AlignmentMatch, 1,64) as AlignmentMatch,
  substr(AlignmentSubSeq, 1,64) as AlignmentSubSeq
FROM hmmpfam_nickname
WHERE HmmQSeq =
  'MTMFWQQNVVDHQSDQDKQAKGAAPT KRLNISFNVKI AVNVNTKMTTTHINQQAPGTSS...';

```

hmmsearch プログラムの場合の例:

この例では、hmmsearch プログラム用の Windows フェデレーテッド・サーバー上にラッパー、サーバー定義、およびニックネームが作成されます。

```

CREATE WRAPPER hmmer_wrapper
  LIBRARY 'db21shmmmer.dll'
  OPTIONS(DB2_FENCED 'Y');

CREATE SERVER hmmsearch_serv
  TYPE search VERSION 2.2
  WRAPPER hmmer_wrapper
  OPTIONS(NODE 'localhost');

CREATE NICKNAME artemia
  FOR SERVER hmmsearch_server
  OPTIONS(DATASOURCE 'artemia', TIMEOUT '1');

SELECT Model, Sequence, substr(SequenceDescription,1,50)
  as SequenceDescription, SequenceScore, SequenceEValue,
  SequenceHits, DomainNumber, SequenceFrom,
  SequenceTo, SequenceFromGlobal, SequenceToGlobal,
  HmmFrom, HmmTo, HmmFromGlobal, HmmToGlobal, DomainScore,
  DomainEValue,
  length(AlignmentConsensus) as "length(AConsensus)",
  length(AlignmentMatch) as "length(AMatch)",
  length(AlignmentSubSeq) as "length(ASubSeq)",
  substr(AlignmentConsensus,1,200) as AlignmentConsensus,
  substr(AlignmentMatch, 1,200) as AlignmentMatch,
  substr(AlignmentSubSeq, 1,200) as AlignmentSubSeq
FROM artemia
WHERE Model = 'globin' and DomainScore > 50;

```

関連タスク:

- 296 ページの『HMMER データ・ソースのニックネームの登録』
- 304 ページの『サンプルを使用した新規 HMMER 照会の構成』
- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』

サンプルを使用した新規 HMMER 照会の構成

以下のサンプル HMMER 照会は、HMMER データ・ソースに対する照会の作成方法を示しています。

手順:

照会を実行するために、ガイドとして以下の例を使用します。

これらの照会において、ニックネームは、HMMER 検索およびデータ・ソースのタイプを記述する名前です。また一部の例は、他のデータ・ソースとともに HMMER ラッパーを使用する方法を示しています。

照会 1

```
SELECT Model, ModelScore, ModelEValue, DomainNumber, DomainScore, DomainEValue
FROM hmmpfam_nickname
WHERE HmmQSeq = 'MTMFWQQNVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQ...'
```

この SQL ステートメントが実行されると、ラッパーは、指示されたシーケンスおよびニックネームによって定義された HMM データベースを使用して、hmmpfam プログラムを実行します。ラッパーは、SELECT ステートメント内でリストされる列を戻します。

照会 2

```
SELECT Model, ModelScore, ModelEValue
FROM hmmpfam_nickname
WHERE HmmQSeq = 'MTMFWQQNVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQ...'
```

この SQL ステートメントが実行されると、ラッパーは、指示された遺伝子列を使用する hmmpfam_nickname の hmmpfam 検索を実行します。加えて、ラッパーは、hmmpfam コマンドに -T 0 オプションを渡します。このオプションは、HMMER ニックネームの固定列のリストにあります。ラッパーは、SELECT の後ろにリストされる 3 つの列を戻します。

照会 3

```
SELECT Model, DomainNumber, DomainScore, DomainEValue
FROM hmmpfam_nickname
WHERE HmmQSeq = 'MTMFWQQNVDHQSDEQDKQAKGAAPT KRLNISFNVKIAVNVNTKMTTTHINQ...'
```

この SQL ステートメントが実行されると、ラッパーは、指示された遺伝子列を使用する hmmpfam_nickname の hmmpfam 検索を実行します。加えて、ラッパーは、hmmpfam コマンドに -E 1 オプションを渡します。このオプションは、HMMER ニックネームの固定列のリストにあります。ラッパーは、SELECT の後ろにリストされる 4 つの列を戻し、DomainScore ごとに高いものから低いものへ順に結果をソートします。

照会 4

```
CREATE WRAPPER hmmer_wrapper
LIBRARY 'db21shmmmer.dll';

CREATE SERVER hmsearch_server
```

```

TYPE search VERSION 2.2
WRAPPER hmmer_wrapper
OPTIONS(NODE 'HMMERserv.MyCompany.com');

CREATE NICKNAME artemia_nickname
FOR SERVER hmmsearch_server
OPTIONS(DATASOURCE 'artemia', TIMEOUT '1');

SELECT Model, Sequence, substr(SequenceDescription,1,50)
as SequenceDescription, SequenceScore, SequenceEValue,
SequenceHits, DomainNumber, SequenceFrom,
SequenceTo, SequenceFromGlobal, SequenceToGlobal,
HmmFrom, HmmTo, HmmFromGlobal, HmmToGlobal, DomainScore,
DomainEValue,
length(AlignmentConsensus) as "length(AConsensus)",
length(AlignmentMatch) as "length(AMatch)",
length(AlignmentSubSeq) as "length(ASubSeq)",
substr(AlignmentConsensus,1,200) as AlignmentConsensus,
substr(AlignmentMatch, 1,200) as AlignmentMatch,
substr(AlignmentSubSeq, 1,200) as AlignmentSubSeq
FROM artemia_nickname
WHERE Model = 'globin' and DomainScore > 50;

```

この SQL ステートメントが実行されると、ラッパーは、globin によって指定された HMM を使用して、シーケンス・ファイル artemia に対して hmmsearch を実行します。DomainScore が 50 以上の行は戻されます。これは、ラッパーが --domT 50 オプションを **hmmsearch** コマンドに渡すためです。ラッパーは SELECT の後に指定された列を戻します。200 文字より長い列値は切り捨てられます。これらの列内の最初の 200 文字だけが戻されます。

関連タスク:

- 284 ページの『フェデレーテッド・サーバーへの HMMER データ・ソースの追加』

関連資料:

- 297 ページの『HMMER ニックネームの固定列』
- 302 ページの『HMMER データ・ソース - 完全指定の例』
- 305 ページの『HMMER ラッパーのメッセージ』

HMMER ラッパーのメッセージ

HMMER のラッパーが機能するには、HmmQSeq 列に述部を含む照会を指定する必要があります。 HmmQSeq 列に述部がないフラグメントを照会すると、エラーが発生します。

このセクションでは、HMMER ラッパーを処理するときに出されるメッセージをリストして説明します。

表 75. HMMER ラッパーのメッセージ

エラー・コード	メッセージ	説明
SQL0142N	SQL ステートメントはサポートされません。	DB2 にサブミットされた SQL 照会はラッパーによって処理できませんでした。必須述部を追加して再サブミットしてください。述部で使用されている演算子はその列に対して有効であるかどうか検査してください。HMMER ニックネームの固定列を参照してください。
SQL1822N	データ・ソース "Hmmer wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「NODE ホスト名を解決できない」です。	CREATE SERVER で指定されている TCP/IP ノード名は無効です。
SQL1822N	データ・ソース "Hmmer wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「デーモンに接続不可」です。	hmmmer_daemon プログラムが現在ターゲット・ノードで稼働していないか、CREATE SERVER コマンドで指定されている DAEMON_PORT がデーモン構成ファイル HMMER_DAEMON.config で指定された DAEMON_PORT 値に一致しないかのどちらかです。
SQL1822N	データ・ソース "Hmmer wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「hmmmer デーモンの不明なエラー」です。	CREATE NICKNAME ステートメントで指定されている DATASOURCE 名は、デーモン構成ファイル HMMER_DAEMON.config にリストされているプロファイル・データベース名のどれにも一致しません。
SQL1822N	データ・ソース "Hmmer wrapper" から予期しないエラー・コード "Unspecified Error" を受け取りました。関連するテキストおよびトークンは「重要: "--cut_TC" オプションは無効」です。	CutMode 述部は小文字で指定する必要があります。例: WHERE CutMode = 'tc'

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 15 章 Informix データ・ソースへのアクセスの構成

この章では、Informix データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Informix データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

フェデレーテッド・サーバーへの Informix の追加

フェデレーテッド・サーバーへの Informix データ・ソースの追加

フェデレーテッド・サーバーが Informix データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供する必要があります。

DB2 コントロール・センター、または DB2 コマンド行を使用して、Informix データ・ソースにアクセスできるようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- フェデレーション用に構成される DB2 サーバー。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- フェデレーテッド・サーバー上にインストールされて構成された Informix クライアント SDK ソフトウェア。
- AIX フェデレーテッド・サーバーでは、AIX Base Application Development Math Library。このライブラリーがインストールされているかどうかは、AIX コマンド **lsllpp -l bos.adt.libm** で確認できます。

手順:

Informix データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. Informix クライアント構成ファイルをセットアップしてテストします。
2. Informix 環境変数を設定します。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Informix サーバーとの接続をテストします。

7. Informix の表、ビュー、およびシノニムのニックネームを登録します。

関連概念:

- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 49 ページの『FEDERATED パラメーターのチェック』
- 57 ページの『フェデレーテッド・データベースの作成』
- 308 ページの『Informix クライアント構成ファイルのセットアップおよびテスト』
- 313 ページの『Informix ラッパーの登録』
- 315 ページの『Informix データ・ソースのサーバー定義の登録』
- 318 ページの『Informix データ・ソースのユーザー・マッピングの作成』
- 319 ページの『Informix サーバーへの接続のテスト』
- 320 ページの『Informix 表、ビュー、およびシノニムのニックネームの登録』
- 322 ページの『Informix データ・ソースに対する構成のチューニングおよびトラブルシューティング』
- 309 ページの『Informix 環境変数の設定』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Informix クライアント構成ファイルのセットアップおよびテスト

Informix クライアント構成ファイルのセットアップとテストは、Informix データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

クライアント構成ファイルは、フェデレーテッド・サーバーにインストールされているクライアント・ライブラリーを使用して、Informix に接続するために使用されます。このファイルは、各 Informix データベース・サーバーのロケーションおよび、そのデータベース・サーバーへの接続のタイプ (プロトコル) を指定します。

クライアント構成ファイルのデフォルトのロケーションは、フェデレーテッド・サーバーによって使用されるオペレーティング・システムによって異なります。

- UNIX では、このファイルのデフォルトのロケーションおよび名前は `$INFORMIXDIR/etc/sqlhosts` です。sqlhosts ファイルは、Informix クライアント SDK でインストールされます。
- Windows では、sqlhosts レジストリーのデフォルトのロケーションはローカル・コンピューターです。

sqlhosts のフォーマットは、「*Administrator's Guide for Informix Dynamic Server*」に記載されています。

手順:

Informix クライアント構成ファイルをセットアップおよびテストするには、次のようにします。

1. Informix クライアント SDK を構成します。
 - UNIX の場合、sqlhosts ファイルを編集して Informix クライアント SDK を構成することができます。sqlhosts ファイルは、Informix Connect または Informix クライアント SDK がインストールされている別のシステムからコピーすることもできます。
 - Windows の場合、Informix Setnet32 ユーティリティを使用して Informix クライアント SDK を構成することができます。Setnet32 ユーティリティは、sqlhosts レジストリーをセットアップします。
2. sqlhosts ファイルまたはレジストリーの位置を検査します。
 - UNIX では、sqlhosts ファイルは \$ INFORMIXDIR/etc/ ディレクトリーにあります。
 - Windows では、sqlhosts 情報は Windows レジストリーの次のキーに保持されます。
HKEY_LOCAL_MACHINE\SOFTWARE\INFORMIX\SQLHOSTS
3. sqlhosts ファイルまたはレジストリーがデフォルトのロケーションにない場合は、環境変数 INFORMIXSQLHOSTS を設定してください。
 - UNIX オペレーティング・システムでは、環境変数 INFORMIXSQLHOSTS を sqlhosts ファイルの完全修飾名に設定します。
 - Windows オペレーティング・システムでは、環境変数 INFORMIXSQLHOSTS を、レジストリーが保管されている Windows コンピューターの名前に設定します。
4. 接続をテストして、クライアント・ソフトウェアが Informix サーバーに接続できることを確認してください。Informix **dbaccess** ツールがフェデレーテッド・サーバーにある場合は、そのツールを使用して接続をテストしてください。ない場合は、Informix デモ・プログラムを実行して、クライアントのセットアップをテストしてください。

この一連の作業における次のタスクは、『Informix 環境変数の設定』です。

関連タスク:

- 313 ページの『Informix ラッパーの登録』
- 322 ページの『Informix データ・ソースに対する構成のチューニングおよびトラブルシューティング』

Informix 環境変数の設定

Informix 環境変数の設定は、フェデレーテッド・サーバーに Informix を追加するという、より大きな作業の一部を成しています。

DB2 Information Integrator をインストールする場合、インストール・プロセスは db2dj.ini ファイル内で Informix 環境変数の設定します。

以下の場合、環境変数は db2dj.ini ファイルに設定されません。

- DB2 フェデレーテッド・サーバーのセットアップ後に Informix クライアント・ソフトウェアをインストールする。
- Informix クライアント・ソフトウェアがインストールされていない。

Informix の有効な環境変数は以下のとおりです。

- INFORMIXDIR
- INFORMIXSERVER
- INFORMIXSQLHOSTS (オプション)
- CLIENT_LOCALE (オプション)
- DB_LOCALE (オプション)
- DBNLS (オプション)

オプション環境変数は手動で設定する必要があります。

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。『db2dj.ini ファイルの制約事項』

手順:

必要な環境変数を自動的に設定するには、次のようにします。

1. DB2 フェデレーテッド・サーバーでクライアント・ソフトウェアをインストールして構成します (必要な場合)。
2. 必要な環境変数を設定します。DB2 サーバーのインストールを再度実行することによって環境変数を自動的に設定し、標準インストール・オプションを指定することができます。

環境変数を手動で設定するには、次のようにします。

1. db2dj.ini ファイルを編集します。
 - Windows を実行しているフェデレーテッド・サーバーの場合、このファイルは %DB2PATH%\cfg ディレクトリ内にあります。
 - UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは sql1lib/cfg ディレクトリ内にあります。

db2dj.ini ファイルには、ご使用のフェデレーテッド・サーバーにインストールされている Informix クライアント・ソフトウェアに関する構成情報が含まれます。このファイルがない場合は、任意のテキスト・エディターを使用して、db2dj.ini という名前の新規ファイルを作成することができます。db2dj.ini ファイルには、変数に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。

2. 必要に応じて以下の環境変数を設定します。

INFORMIXDIR

INFORMIXDIR 環境変数を、Informix クライアント SDK ソフトウェアがインストールされているディレクトリー・パスに設定します。例えば、以下のようになります。

Windows を実行しているフェデレーテッド・サーバーの場合、パスを次のように設定します。

```
INFORMIXDIR=C:\informix\csdk
```

UNIX を実行しているフェデレーテッド・サーバーの場合、パスを次のように設定します。

```
INFORMIXDIR=/informix/csdk
```

INFORMIXSERVER

この変数は、デフォルトの Informix サーバーの名前を識別します。この設定は、sqlhosts ファイル (UNIX) または SQLHOSTS レジストリー・キー (Windows) で有効な項目でなければなりません。

INFORMIXSERVER の値を入手するには、sqlhosts ファイルを読み取ります。dbservername 値のいずれかを選択します。dbservername は、sqlhosts 内の各項目の最初の値です。例えば、以下のようになります。

```
INFORMIXSERVER=inf93
```

要件: Informix ラッパーがこの変数の値を使用しない場合でも、Informix クライアントによってこの変数を設定するよう要求されます。ラッパーは、アクセスしたい Informix データベース・サーバーを指定する

NOTE サーバー・オプションの値を使用します。

INFORMIXSQLHOSTS

Informix sqlhosts ファイルのデフォルトのパスを使用している場合、この変数を設定する必要はありません。ただし、Informix sqlhosts ファイルの他のパスを使用している場合、この変数を Informix sqlhosts ファイルがある絶対パス名に設定する必要があります。

- UNIX を実行しているフェデレーテッド・サーバーの場合、デフォルトのパスは \$INFORMIXDIR/etc です。
- Windows を実行しているフェデレーテッド・サーバーの場合、SQLHOSTS レジストリー・キーがローカル・コンピューター上にない場合には、レジストリーを保管する Windows コンピューターの名前は INFORMIXSQLHOSTS になります。

この変数を別のパスに設定する UNIX の例は、次のとおりです。

```
INFORMIXSQLHOSTS=/informix/csdk/etc/my_sqlhosts
```

3. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。DB2 インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop  
db2start
```

Informix コード・ページ変換のセットアップ:

Informix コード・ページ変換の場合、以下のオプション環境変数を設定することができます。

- CLIENT_LOCALE

- DB_LOCALE
- DBNLS

Informix ラッパーが Informix データ・ソースに接続するたび、ラッパーは、その接続に使用するコード・ページ値を判別します。Informix 環境変数 `CLIENT_LOCALE` がフェデレーテッド・サーバー上の `db2dj.ini` ファイルに設定されている場合、ラッパーは `db2dj.ini` ファイルの値を使用します。

Informix サーバー上で `glfiles` コマンドを発行することによって、有効な Informix ロケールのリストを入手できます。コード・ページ変換についての詳細は、『*Informix Guide to GLS Functionality*』を参照してください。

Informix コード・ページ環境変数は以下のとおりです。

CLIENT_LOCALE

`CLIENT_LOCALE` 環境変数を使用したい Informix ロケールに設定します。`CLIENT_LOCALE` を設定しなかった場合、ラッパーは、フェデレーテッド・データベースのコード・ページおよびテリトリーを判別します。ラッパーは、`CLIENT_LOCALE` 変数を最もよく一致する Informix ロケールに設定します。一致する Informix ロケールがない場合、ラッパーは、`CLIENT_LOCALE` 変数を UNIX システムの場合には `en_us.8859-1` ロケールに、Windows システムの場合には `en_us.CP1252` ロケールに設定します。Informix `glfiles` コマンドを使用して、ロケール名のリストを参照することができます。

```
CLIENT_LOCALE=Informix_client_locale_value
```

DB_LOCALE

Informix データベースが、クライアント・ロケールとは異なるコード・ページを使用しており、Informix が 2 つのコード・ページ間で変換を実行するようにしたい場合に、この環境変数を設定します。Informix 環境変数 `DB_LOCALE` を Informix データベース・ロケールの名前に設定します。たとえば、以下のようにします。

```
DB_LOCALE=Informix_db_locale_value
```

DBNLS

`DB_LOCALE` 設定が Informix データベースの実際のロケールと一致することを Informix が検証できるようにするには、この Informix 環境変数を 1 に設定します。

```
DBNLS=1
```

コード・ページ環境変数 - 例:

Informix データベースが、クライアント・ロケールとは異なるコード・ページを使用しており、Informix が 2 つのコード・ページ間で変換を実行するようにしたいとします。Informix 環境変数 `DB_LOCALE` を Informix データベース・ロケールの名前に設定する必要があります。フェデレーテッド・サーバー上の `db2dj.ini` ファイル内で、この変数を設定します。

DB_LOCALE 設定が Informix データベースの実際のロケールと一致することを Informix が検証できるようにするには、Informix 環境変数 DBNLS を 1 に設定する必要があります。この変数は、フェデレーテッド・サーバー上の db2dj.ini ファイル内で設定します。

中国語コード・ページ GB 18030 を使用するデータを含むデータ・ソースにアクセスする場合、フェデレーテッド・データベースは UTF-8 コード・ページを使用する必要があります。Informix ラッパーは Informix 環境変数を以下のように設定します。

```
CLIENT_LOCALE=zh_cn.UTF8  
GL_USEGLU=1
```

以下の設定を db2dj.ini ファイルに追加して、Informix クライアントが GB 18030 データを Unicode に正しく変換できるようにする必要があります。

```
DB_LOCALE=zh_cn.GB18030-2000
```

この一連の作業における次のタスクは、『Informix ラッパーの登録』です。

関連タスク:

- 320 ページの『Informix 表、ビュー、およびシノニムのニックネームの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』
- 66 ページの『db2dj.ini ファイルの制約事項』

Informix ラッパーの登録

Informix ラッパーの登録は、フェデレーテッド・サーバーに Informix データ・ソースを追加するという、より大きな作業の一部を成しています。

Informix データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER INFORMIX
```

推奨事項: INFORMIX というデフォルトのラッパー名を使用してください。デフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパー名に関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは異なる名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターも含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で `inf_wrapper` という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER inf_wrapper LIBRARY 'libdb2informix.a'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『Informix ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Informix ラッパーのサーバー定義の登録』です。

関連タスク:

- 315 ページの『Informix データ・ソースのサーバー定義の登録』

関連資料:

- 314 ページの『Informix ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Informix ラッパー・ライブラリー・ファイル

次の表には、Informix ラッパーのディレクトリー・パスおよびライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、`libdb2informix.a`、`libdb2informixF.a`、および `libdb2informixU.a` です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 76. Informix ラッパー・ライブラリー・ロケーションおよびファイル名

オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
AIX	<code>/usr/opt/db2_08_01/lib/</code>	<code>libdb2informix.a</code>
HP-UX	<code>/opt/IBM/db2/V8.1/lib</code>	<code>libdb2informix.sl</code>
Linux	<code>/opt/IBM/db2/V8.1/lib</code>	<code>libdb2informix.so</code>
Solaris	<code>/opt/IBM/db2/V8.1/lib</code>	<code>libdb2informix.so</code>
Windows	<code>%DB2PATH%¥bin</code>	<code>db2informix.dll</code>

`%DB2PATH%` は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、`C:¥Program Files¥IBM¥SQLLIB` です。

関連タスク:

- 313 ページの『Informix ラッパーの登録』

Informix データ・ソースのサーバー定義の登録

Informix データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに Informix データ・ソースを追加するという、より大きな作業の一部を成しています。

アクセスする各 Informix サーバーを、フェデレーテッド・データベース内に定義する必要があります。

手順:

サーバー定義は、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- このタスクを DB2 コントロール・センターから実行するには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、あるいは「サーバー定義 (Server Definitions)」フォルダーを右マウス・ボタンでクリックして、「作成」をクリックします。Discover ツールは、Informix サーバーのノード名を検索します。サーバー定義を登録するには、DBNAME サーバー・オプションの情報を指定する必要があります。
- このタスクを DB2 コマンド行から行うには、CREATE SERVER ステートメントを使用します。

1. Informix sqlhosts ファイルまたはレジストリーでノード名を見つけます。

sqlhosts ファイルの例:

```
inf724 onsoctcp anaconda inmx724
inf731 onsoctcp boa ifmx731
inf92 onsoctcp python ifmx92
```

各行の最初の値は inf724 などの *node_name* です。

各行の 2 番目の値は *nettype*、または接続のタイプです。この例の場合、onsoctcp はこれが TCP/IP 接続であることを示します。

各行の 3 番目の値は anaconda、boa、python などのホスト名です。

各行の 4 番目の値は inmx724 などのサービス名です。サービス名フィールドは、2 番目の値にリストされた *nettype* により異なります。

node_name は CREATE SERVER SQL ステートメントでオプションとなっていますが、Informix データ・ソースの場合は必要です。

このファイルのフォーマットおよび、それらのフィールドの意味の詳細は、Informix 資料「Administrators Guide for Informix Dynamic Server」を参照してください。

2. CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE informix
VERSION 9 WRAPPER INFORMIX
OPTIONS (NODE 'node_name', DBNAME 'db_name');
```

サーバー定義を登録した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

この一連の作業における次のタスクは、『Informix データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 318 ページの『Informix データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 316 ページの『CREATE SERVER ステートメント - Informix ラッパーの例』

CREATE SERVER ステートメント - Informix ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して Informix ラッパー用のサーバー定義を登録する方法を示すいくつかの例が提供されます。必要パラメーターを使用してサーバー定義を登録する方法の完全指定の例、およびその他のサーバー・オプションの例がこのトピックには含まれています。

完全指定の例:

以下の例は、CREATE SERVER ステートメントを使用して、Informix ラッパーのサーバー定義を登録する方法を示しています。

```
CREATE SERVER asia TYPE informix VERSION 9 WRAPPER INFORMIX
OPTIONS (NODE 'abc', DBNAME 'sales', IUD_APP_SVPT_ENFORCE 'N')
```

asia Informix データベース・サーバーに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE *informix*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。Informix ラッパーの場合、サーバー・タイプは *informix* でなければなりません。

VERSION 9

アクセスする Informix データベース・サーバーのバージョン。サポートされる Informix バージョンは 7、8、および 9 です。

WRAPPER *INFORMIX*

CREATE WRAPPER ステートメントで指定した名前。

NODE 'abc'

Informix データベース・サーバーが存在するノードの名前。ノード名は `sqlhosts` ファイルから入手します。この値は大文字小文字の区別がありません。

ノード名は CREATE SERVER ステートメントでオプションとなっていますが、Informix データ・ソースの場合は必要です。

DBNAME 'sales'

これは、アクセスしたい Informix データベースの名前です。この値は大文字小文字の区別があります。

データベース名は CREATE SERVER ステートメントでオプションとなっていますが、Informix データ・ソースの場合は必要です。

IUD_APP_SVPT_ENFORCE 'N'

DB2 フェデレーテッド・システムが、アプリケーション・セーブポイント・ステートメントの検出または作成を実施すべきかどうかを指定します。Informix はアプリケーション保管点ステートメントをサポートしません。'N' にすると、エラーが検出されても、フェデレーテッド・サーバーはトランザクションをロールバックしません。エラー・リカバリーは、アプリケーションが処理する必要があります。

IUD_APP_SVPT_ENFORCE サーバー・オプションは、Informix データ・ソースとの間で複製が行えるよう、'N' に設定する必要があります。

サーバー・オプションの例:

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと Informix に特有のサーバー・オプションを指定できます。

以下の例は、サーバー・オプションを追加した Informix サーバー定義を示します。

```
CREATE SERVER asia TYPE informix VERSION 9 WRAPPER INFORMIX
  OPTIONS (NODE 'abc', DBNAME 'sales', FOLD_ID 'N', FOLD_PW 'N')
```

データ・ソースに接続する時に、フェデレーテッド・サーバーは、ユーザー ID とパスワードについて、入力された文字だけでなく、大文字小文字のすべての可能な組み合わせを使用して接続を試みます。フェデレーテッド・サーバーはデータ・ソース・サーバーとの接続に成功するまで、最高 9 回、接続を試みます。このような試みは接続時間を長引かせることになり、ユーザー ID がロックアウトされる結果になる可能性もあります。FOLD_ID および FOLD_PW サーバー・オプションに値を指定すれば、ロックアウトを防ぐことができます。

たとえば、FOLD_ID と FOLD_PW のサーバー・オプションを 'N' (ユーザー ID またはパスワードを大文字変換しない) に設定することができます。このような設定をした場合は、大文字小文字を正しく使用して、ユーザー ID とパスワードを指定する必要があります。これらのオプションを 'N' にすると、誤ったユーザー ID やパスワードが指定された場合に、ラッパーがさまざまな大文字小文字の組み合わせを試行しないで済みます。これらの 2 つのサーバー・オプションにより、ログインに何回も失敗して限度を超え、ID がロックアウトされる可能性を減らすことができます。

関連タスク:

- 315 ページの『Informix データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

Informix データ・ソースのユーザー・マッピングの作成

Informix データ・ソースのユーザー・マッピングの作成は、Informix データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Informix サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Informix データ・ソースへの分散要求を送信します。

手順:

ローカル・ユーザー ID を Informix サーバーのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR USERID SERVER INFORMIXSERVER
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

この一連の作業における次のタスクは、『Informix サーバーへの接続のテスト』です。

関連タスク:

- 319 ページの『Informix サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 318 ページの『CREATE USER MAPPING ステートメント - Informix ラッパーの例』

CREATE USER MAPPING ステートメント - Informix ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Informix サーバーのユーザー ID およびパスワードにマップする方法を示す例が提供されます。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下の例は、フェデレーテッド・サーバーのユーザー ID (VINCENT) を Informix サーバーのユーザー ID およびパスワード ('vinnie' および 'close2call') にマップする方法を示しています。

```
CREATE USER MAPPING FOR VINCENT SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call')
```

VINCENT

Informix サーバーで定義されるユーザー ID にマップする、ローカルのユーザー ID を指定します。

SERVER asia

CREATE SERVER ステートメントで登録した Informix サーバーの名前を指定します。

REMOTE_AUTHID 'vinnie'

VINCENT をマップする、Informix データベース・サーバー側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

REMOTE_PASSWORD 'close2call'

'vinnie' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER asia
  OPTIONS (REMOTE_AUTHID 'vinnie', REMOTE_PASSWORD 'close2call')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 318 ページの『Informix データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

Informix サーバーへの接続のテスト

Informix サーバーへの接続のテストは、Informix データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

サーバー定義および定義したユーザー・マッピングを使用して、Informix サーバーへの接続をテストできます。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、Informix システム表に対して SQL SELECT ステートメントを発行します。

例:

```
SET PASSTHRU server_name
SELECT count(*) FROM informix.systables
SET PASSTHRU RESET
```

SQL SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - Informix サーバーを調べ、入ってくる接続に対して構成されていることを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD のオプションが Informix サーバーとの接続に有効な設定になっているかを確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の Informix クライアント SDK ソフトウェアを調べ、これが正しくインストールされ、Informix サーバーに接続するように構成されていることを確認します。
 - DB2 フェデレーテッド変数の設定を調べ、Informix サーバーに合わせた設定になっているかどうかを確認します。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
 - サーバー定義を調べます。必要であれば、それをドロップして再作成します。

この一連の作業における次のタスクは、『Informix 表、ビュー、およびシノニムのニックネームの登録』です。

関連タスク:

- 307 ページの『フェデレーテッド・サーバーへの Informix データ・ソースの追加』
- 320 ページの『Informix 表、ビュー、およびシノニムのニックネームの登録』
- 309 ページの『Informix 環境変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

Informix 表、ビュー、およびシノニムのニックネームの登録

Informix 表、ビュー、およびシノニムのニックネームの登録は、フェデレーテッド・サーバーに Informix データ・ソースを追加するという、より大きな作業の一部を成しています。

定義するそれぞれの Informix サーバーで、アクセスしたい各表、ビュー、またはシノニムごとにニックネームを登録します。Informix サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、**CREATE NICKNAME** ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録した時に収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れられます。オプティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用するため、ニックネームを登録する前に、データ・ソース側で (**DB2** の **RUNSTATS** と同等のデータ・ソース・コマンドを使用して) 統計を更新してください。

手順:

ニックネームを登録するには、**CREATE NICKNAME** ステートメントを発行します。

例:

```
CREATE NICKNAME informix_name FOR INFOSERVER."remote_schema"."remote.table"
```

ニックネームの長さは 128 文字までです。

このステップを、ニックネームを作成したいそれぞれの Informix 表、ビュー、またはシノニムごとに繰り返します。

ニックネームを作成すると、DB2 は接続を使用してデータ・ソース・カタログを照会します。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連資料:

- 「コマンド・リファレンス」の『**RUNSTATS** コマンド』
- 「SQL リファレンス 第 2 巻」の『**CREATE NICKNAME** ステートメント』
- 321 ページの『**CREATE NICKNAME** ステートメント - Informix ラッパーの例』

CREATE NICKNAME ステートメント - Informix ラッパーの例

このトピックでは、**CREATE NICKNAME** ステートメントを使用して、アクセスしたい Informix の表、ビュー、またはシノニムのニックネームを登録する方法の例が提供されます。

この例では、ニックネームが割り当てられる Informix サーバーのリモート・オブジェクトを指定する方法が示されます。

```
CREATE NICKNAME JPSALES FOR asia."salesdata"."japan"
```

JPSALES

Informix の表、ビュー、またはシノニムを識別するために使用する固有なニックネーム。

注: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

asia."salesdata"."japan"

リモート・オブジェクトを表す、3つの部分からなる ID。

- *asia* は、CREATE SERVER ステートメントで Informix データベース・サーバーに割り当てた名前です。
- *salesdata* は、表、ビュー、またはシノニムが属するリモート・スキーマの名前です。
- *japan* は、アクセスしたいリモートの表、ビュー、またはシノニムの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Informix スキーマおよび表の名前を大文字に変換します。

関連タスク:

- 320 ページの『Informix 表、ビュー、およびシノニムのニックネームの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE NICKNAME ステートメント』

Informix データ・ソースに対する構成のチューニングおよびトラブルシューティング

Informix データ・ソースに対する構成をセットアップした後、パフォーマンスを改善するために構成を変更したい場合があります。

FOLD_ID および FOLD_PW サーバー・オプションを設定してパフォーマンスを改善する

データ・ソースに接続する時に、サーバーは、ユーザー ID とパスワードについて、大文字小文字のすべての可能な組み合わせを使用して接続を試みます。サーバーはデータ・ソース・サーバーとの接続に成功するまで、最高 9 回、接続を試みます。このような試みは接続時間を長引かせることになり、ユーザー ID がロックアウトされる結果になる可能性もあります。

手順:

パフォーマンスを改善するためには、ALTER SERVER OPTION ステートメントを使用して、FOLD_ID および FOLD_PW サーバー・オプションの値を指定します。

- Informix のすべてのユーザー ID とパスワードが小文字である場合、FOLD_ID および FOLD_PW サーバー・オプションを値 'L' を使用して設定することにより、接続時間が改善されます。例:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'L');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'L');
```

- フェデレーテッド・サーバーは、ユーザー ID およびパスワードの大文字小文字の値のそれぞれの組み合わせを試みます。これらのオプションを 'N' (ユーザー ID とパスワードを大文字に変換しない) にすれば、何回もログインを試みて失敗し、失敗ログインの最大数を越えるという可能性を減らすことができます。このような設定をした場合は、常に大文字小文字を正しく使用して、ユーザー ID と

パスワードを指定する必要があります。誤ったユーザー ID とパスワードが指定されても、ラッパーは組み合わせを変えて試行することはありません。例:

```
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_ID 'N');
ALTER SERVER TYPE INFORMIX
  OPTIONS (ADD FOLD_PW 'N');
```

関連タスク:

- 307 ページの『フェデレーテッド・サーバーへの Informix データ・ソースの追加』

関連資料:

- 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』
- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』

第 16 章 Microsoft SQL Server データ・ソースへのアクセスの構成

この章では、Microsoft SQL Server データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Microsoft SQL Server データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要のある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

フェデレーテッド・サーバーへの Microsoft SQL Server の追加

フェデレーテッド・サーバーへの Microsoft SQL Server データ・ソースの追加

Microsoft SQL Server データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスしたいデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供しなければなりません。

Microsoft SQL Server データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、DB2 コントロール・センターか DB2 コマンド行を使用することができます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- ODBC ドライバーがフェデレーテッド・サーバー上にインストール・構成されていなければなりません。サポートされるドライバーは Microsoft ODBC ドライバー (Windows) および DataDirect Technologies Connect for ODBC ドライバー (UNIX) です。

手順:

フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するには、以下のようにします。

1. フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します。
 - Windows では、ODBC システム DSN が適切にセットアップされていることを確認し、Microsoft SQL Server リモート・サーバーへの接続をテストします。

- UNIX システムでは、odbc.ini ファイルを更新または作成し、Microsoft SQL Server リモート・サーバーへの接続をテストします。
- 2. Microsoft SQL Server ラッパー用の環境変数を設定します。
- 3. ラッパーを登録します。
- 4. サーバー定義を登録します。
- 5. ユーザー・マッピングを作成します。
- 6. Microsoft SQL Server リモート・サーバーへの接続をテストします。
- 7. Microsoft SQL Server の表およびビューのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 326 ページの『Microsoft SQL Server データ・ソースにアクセスするためのフェデレーテッド・サーバーの準備』
- 330 ページの『Microsoft SQL Server ラッパーの登録』
- 332 ページの『Microsoft SQL Server データ・ソースのサーバー定義の登録』
- 335 ページの『Microsoft SQL Server データ・ソースのユーザー・マッピングの作成』
- 336 ページの『Microsoft SQL Server リモート・サーバーへの接続のテスト』
- 338 ページの『Microsoft SQL Server 表およびビューのニックネームの登録』
- 339 ページの『Microsoft SQL Server データ・ソースのための構成のチューニングとトラブルシューティング』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Microsoft SQL Server データ・ソースにアクセスするためのフェデレーテッド・サーバーの準備

Microsoft SQL Server データ・ソースにアクセスさせるためのフェデレーテッド・サーバーとデータベースの準備は、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

Microsoft SQL Server データ・ソースにアクセスさせるためにフェデレーテッド・サーバーとデータベースを準備するためのステップは、フェデレーテッド・サーバーで稼働しているオペレーティング・システムによって異なります。

手順:

フェデレーテッド・サーバーの準備は、以下のように行います。

Windows の場合:

1. 「コントロール パネル」で設定をチェックして、ODBC システム DSN が Microsoft SQL Server データ・ソースに接続するように設定されていることを確認します。Microsoft SQL Server リモート・サーバーの既存の項目を見つけるか、新規の項目を作成します。この項目は、フェデレーテッド・データベースにサーバーを登録するときに NODE サーバー・オプションに使用する値です。
2. 「ODBC データ ソース アドミニストレータ」ウィンドウから、「構成」を選択し、ODBC システム DSN から Microsoft SQL Server データ・ソースへの接続をテストします。あるいは、Microsoft SQL Server の照会ツールを使用して、接続をテストすることもできます。

UNIX の場合:

1. フェデレーテッド・サーバー上の `odbc.ini` ファイルが更新されている (または必要な場合、作成されている) ことを確認してください。

推奨: `odbc.ini` ファイルやこのファイルのコピーは、DB2 インスタンス所有者のホーム・ディレクトリーに置きます。

2. `odbc.ini` へのパスが `ODBCINI` 環境変数にあることを確認します。オペレーティング・システムのコマンド・プロンプトから次のコマンドを発行します。

```
export ODBCINI=$HOME/.odbc.ini
```

3. 適切なシンボリック・リンクを作成します。

- HP-UX の場合、次のシンボリック・リンクを作成する必要があります。

```
ln -s $DJX_ODBC_LIBRARY_PATH/libodbcinst.sl /usr/exe/libodbcinst.sl
```

DataDirect Technologies Connect for ODBC 4.2 ドライバーを使用している場合、次のシンボリック・リンクも作成しなければなりません。

```
ln -s $DJX_ODBC_LIBRARY_PATH/libivicu19.sl /ivicu/exe/libivicu19.sl
```

- Linux の場合、以下のシンボリック・リンクを作成する必要があります。

```
ln -s $DJX_ODBC_LIBRARY_PATH/./locale /usr/local/locale
```

```
ln -s $DJX_ODBC_LIBRARY_PATH/libodbcinst.so /usr/lib/libodbcinst.so
```

DataDirect Technologies Connect for ODBC 4.2 ドライバーを使用している場合、次のシンボリック・リンクも作成しなければなりません。

```
ln -s $DJX_ODBC_LIBRARY_PATH/libivicu19.so /usr/lib/libivicu19.so
```

- Solaris の場合、次のシンボリック・リンクを作成する必要があります。

```
ln -s $DJX_ODBC_LIBRARY_PATH/./locale $HOME/sql1lib/locale
```

`$HOME` は、DB2 インスタンス所有者のホーム・ディレクトリーです。

4. DataDirect Connect ODBC **demoodbc** ツールを使用して、フェデレーテッド・サーバーから Microsoft SQL Server データ・ソースへの接続をテストします。
 - a. **/opt/odbc/odbc.sh** スクリプトを実行します。このスクリプトは、いくつかのオペレーティング・システム特有の環境変数をセットアップします。
 - b. DataDirect Connect ODBC **demoodbc** ツールを使用して、Microsoft SQL Server データ・ソースへの接続をテストします。**demoodbc** ツールは、Connect ODBC ライブラリーの `/demo` サブディレクトリーにあります。

この一連のタスクの次のタスクは、『Microsoft SQL server 環境変数の設定』です。

関連タスク:

- 330 ページの『Microsoft SQL Server ラッパーの登録』

Microsoft SQL Server 環境変数の設定

Microsoft SQL Server 環境変数の設定は、フェデレーテッド・サーバーに Microsoft SQL Server を追加するという、より大きな作業の一部を成しています。

DB2 Information Integrator のインストール時に、インストール処理は db2dj.ini ファイルに Microsoft SQL Server 環境変数を設定しようとしています。

DB2 Information Integrator をインストールした後で Microsoft SQL Server ODBC ドライバーをインストールすると、環境変数は db2dj.ini ファイルに設定されません。

Microsoft SQL Server 用に有効な環境変数は以下のとおりです。

- DJX_ODBC_LIBRARY_PATH
- ODBCINI
- LD_LIBRARY_PATH (Solaris のみ)
- SHLIB_PATH (HP-UX のみ)
- DB2LIBPATH
- DB2ENVLIST

フェデレーテッド・サーバーで HP-UX が稼働し、複数区画のインスタンス構成である場合、userprofile の SHLIB_PATH の値をエクスポートしなければなりません。

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。『db2dj.ini ファイルの制約事項』

手順:

Microsoft SQL Server ODBC ドライバーの環境変数を自動的に設定するには、以下のようになります。

1. Microsoft SQL Server ODBC ドライバーがまだ DB2 フェデレーテッド・サーバーにインストールされていない場合は、インストールします。
2. 必要な環境変数を設定します。DB2 Information Integrator のインストールを再実行することによって、環境変数を自動的に設定できます。ランチパッドから、「製品のインストール」をクリックし、ウィザードの指示に従います。

Microsoft SQL Server ODBC ドライバーの環境変数を手作業で設定するには、以下のようになります。

1. db2dj.ini ファイルを編集します。

- Windows が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib¥cfg` ディレクトリーにあります。
- UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib/cfg` ディレクトリーにあります。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Microsoft SQL Server ODBC ドライバーに関する構成情報が含まれます。このファイルが存在しない場合、任意のテキスト・エディターを使用して、db2dj.ini という名前の新規ファイルを作成できます。db2dj.ini ファイルには、変数に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。

2. 以下の環境変数を (必要に応じて) 設定します。

DJX_ODBC_LIBRARY_PATH

ODBC ライブラリー・ファイルへのディレクトリー・パスを設定します。例えば、以下のようになります。

```
DJX_ODBC_LIBRARY_PATH=ODBC_driver_directory/lib
```

ODBC_driver_directory は、ODBC ドライバーがインストールされているディレクトリー・パスです。

ODBCINI

ODBCINI 環境変数に ODBC 構成ファイル (odbc.ini) があるディレクトリー・パスを設定します。ODBCINI 環境変数はシステム変数として設定してはなりません。例えば、以下のようになります。

```
ODBCINI=/home/db2inst1/.odbc.ini
```

LD_LIBRARY_PATH

Solaris で ODBC ライブラリー・ファイルへのディレクトリー・パスを設定します。例えば、以下のようになります。

```
LD_LIBRARY_PATH=ODBC_driver_directory/lib
```

SHLIB_PATH

HP-UX で ODBC ライブラリー・ファイルへのディレクトリー・パスを設定します。例えば、以下のようになります。

```
SHLIB_PATH=ODBC_driver_directory/lib
```

3. Microsoft SQL Server にアクセスするには、lib サブディレクトリーの ODBC ライブラリー・ファイルへのディレクトリー・パスを設定する必要があります。例えば、以下のようになります。

```
db2set DB2LIBPATH=ODBC_driver_directory/lib
```

4. Connect ODBC ドライバーを使用して Microsoft SQL Server データ・ソースにアクセスするには、DB2ENVLIST に LIBPATH という値を設定します。例えば、以下のようになります。

```
db2set DB2ENVLIST=LIBPATH
```

LIBPATH は、ODBC ドライバーがインストールされているディレクトリー・パスです。

5. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。DB2 インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop  
db2start
```

HP-UX 複数パーティション・インスタンス構成の場合:

1. userprofile がまだないなら、作成します。userprofile は \$HOME/sql/lib/ ディレクトリーにあります。
2. SHLIB_PATH を userprofile に追加します。SHLIB_PATH の値は、ODBC ドライバーがインストールされているディレクトリー・パスです。たとえば、次のコマンドを発行します。

```
export SHLIB_PATH=$SHLIB_PATH:/home/DataDirectODBC/lib
```

この一連のタスクで次のタスクは『Microsoft SQL Server ラッパーの登録』です。

関連タスク:

- 338 ページの『Microsoft SQL Server 表およびビューのニックネームの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』
- 66 ページの『db2dj.ini ファイルの制約事項』

Microsoft SQL Server ラッパーの登録

Microsoft SQL Server ラッパーの登録は、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

Microsoft SQL Server データ・ソースにアクセスするためのラッパーを登録しなければなりません。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行します。

例えば、以下のようになります。

```
CREATE WRAPPER MSSQLODBC3
```

推奨: MSSQLODBC3 というデフォルトのラッパー名を使用してください。このデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパー名に関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォ

ルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で sqlserver_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER sqlserver_wrapper LIBRARY 'libdb2mssql3.a'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『Microsoft SQL Server ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連のタスクの次のタスクは、『Microsoft SQL Server ラッパーのサーバー定義の登録』です。

関連タスク:

- 332 ページの『Microsoft SQL Server データ・ソースのサーバー定義の登録』

関連資料:

- 331 ページの『Microsoft SQL Server ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Microsoft SQL Server ラッパー・ライブラリー・ファイル

次の表は、Microsoft SQL Server ラッパーのためのディレクトリー・パスとライブラリー・ファイル名をリストします。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2mssql3.a、libdb2mssql3F.a、および libdb2mssql3U.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 77. Microsoft SQL Server クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2mssql3.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2mssql3.sl
Linux	/opt/IBM/db2/V8.1/lib	libdb2mssql3.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2mssql3.so
Windows	%DB2PATH%\bin	db2mssql3.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 330 ページの『Microsoft SQL Server ラッパーの登録』

Microsoft SQL Server データ・ソースのサーバー定義の登録

Microsoft SQL Server データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスする各 Microsoft SQL Server リモート・サーバーを定義する必要があります。最初に Microsoft SQL Server リモート・サーバーのノード名を見つけてから、CREATE SERVER ステートメントを発行してサーバー定義を登録するのにこのノード名を使用します。

手順:

Microsoft SQL Server データ・ソースのサーバー定義を登録するには、次のようにします。

1. ノード名を見つけてみます。
 - フェデレーテッド・サーバーが Windows を使用している場合、NODE は、アクセスする Microsoft SQL Server リモート・サーバーに指定したシステム DSN 名です。
 - フェデレーテッド・サーバーが UNIX を使用している場合、NODE は .odbc.ini ファイルに指定します。

以下は AIX 上の .odbc.ini ファイルの例です。

例: AIX の .odbc.ini ファイル

```
rawilson=MS SQL Server 7.0
medusa=MS SQL Server 7.0
[rawilson]
Driver=/opt/odbc/lib/ivmsss16.so
Description=MS SQL Server Driver for AIX
Address=9.112.30.39,1433
[medusa]
Driver=/opt/odbc/lib/ivmsss16.so
Description=MS SQL Server Driver for AIX
Address=9.112.98.123,1433
```

.odbc.ini ファイルの先頭に [ODBC Data Sources] とラベルの付いたセクションがあり、そこにノードがリストされています。ノードごとに、各ノードを記述するセクション [node_name] があります。

ノード名は CREATE SERVER ステートメントでオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

2. CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE MSSQLSERVER VERSION 7.0 WRAPPER mssqlodb3
OPTIONS (NODE 'sqlnode', DBNAME 'mssdb');
```

サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

この一連のタスクの次のタスクは、『Microsoft SQL Server データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 335 ページの『Microsoft SQL Server データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 333 ページの『CREATE SERVER ステートメント - Microsoft SQL Server ラッパーの例』

CREATE SERVER ステートメント - Microsoft SQL Server ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して Microsoft SQL Server ラッパー用にサーバーを登録する方法の例が提供されます。必要パラメーターを使用してサーバーを登録する方法の完全指定の例、およびその他のサーバー・オプションの例がこのトピックには含まれています。

完全指定の例:

以下の例は、CREATE SERVER ステートメントを発行して Microsoft SQL の Server ラッパーのサーバー定義を登録する方法を示しています。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER mssqlodbc3  
    OPTIONS (NODE 'sqlnode', DBNAME 'africa');
```

sqlserver

Microsoft SQL Server リモート・サーバーに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE MSSQLSERVER

アクセスを構成しているデータ・ソースのタイプ。Microsoft SQL Server ラッパー用の TYPE パラメーターは、MSSQLSERVER でなければなりません。

VERSION 7.0

アクセスしたい Microsoft SQL Server データベース・サーバーのソフトウェアのバージョン。サポートされるバージョンは 6.5、7.0 および 2000 です。

WRAPPER mssqlodbc3

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE 'sqlnode'

Windows の場合、アクセスする Microsoft SQL server リモート・サーバー

に指定したシステム DSN 名です。UNIX の場合、.odbc.ini ファイルに定義されたノードです。この値は大文字小文字の区別があります。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

DBNAME 'africa'

アクセスしたいデータベースの名前。この値は大文字小文字の区別がありません。

データベース名は CREATE SERVER ステートメントではオプションとなっていますが、Microsoft SQL Server データ・ソースの場合は必要です。

サーバー・オプションの例:

サーバーを登録する際に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと Microsoft SQL Server に特有のサーバー・オプションがあります。

以下に示すのは、COLLATING_SEQUENCE サーバー・オプションの使用法の例です。

```
CREATE SERVER sqlserver TYPE MSSQLSERVER
  VERSION 7.0
  WRAPPER mssqlodbc3
  OPTIONS (NODE 'sqlnode', DBNAME 'africa', COLLATING_SEQUENCE 'I');
```

COLLATING_SEQUENCE サーバー・オプションは、データ・ソースがフェデレーテッド・サーバーと同じ照合シーケンスを使用しているかどうかを指定します。Windows NT または Windows 2000 で稼働する Microsoft SQL Server データベース・サーバーでは、デフォルトの照合シーケンスは大文字小文字を区別しません (例: 'STEWART' と 'StewART' は等しいと見なされます)。フェデレーテッド・サーバーからの正しい結果を保証するには、COLLATING_SEQUENCE サーバー・オプションを 'I' にしてください。この設定は、Microsoft SQL Server データ・ソースで大文字小文字を区別しないことを指示しています。

データ・ソースから戻される結果がフェデレーテッド・サーバーで照会を処理するときに戻される結果と異なる場合、フェデレーテッド・サーバーは照会をプッシュダウンしません。COLLATING_SEQUENCE サーバー・オプションを 'I' にすると、フェデレーテッド・サーバーは、ストリング・データまたは式があり、以下の文節、述部、または関数を含む照会をプッシュダウンしません。

- GROUP BY 文節
- DISTINCT 文節
- 基本述部 (例: 等しい (=))
- 集約関数 (例: MIN や MAX)

関連タスク:

- 332 ページの『Microsoft SQL Server データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

Microsoft SQL Server データ・ソースのユーザー・マッピングの作成

Microsoft SQL Server データ・ソースのユーザー・マッピングの作成は、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

Microsoft SQL Server データ・ソースにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ローカル・ユーザー ID を Microsoft SQL Server リモート・サーバーのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR userid SERVER sqlserver
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

この一連のタスクの次のタスクは、『Microsoft SQL Server リモート・サーバーへの接続のテスト』です。

関連タスク:

- 336 ページの『Microsoft SQL Server リモート・サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 335 ページの『CREATE USER MAPPING ステートメント - Microsoft SQL Server ラッパーの例』

CREATE USER MAPPING ステートメント - Microsoft SQL Server ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Microsoft SQL server リモート・サーバーのユーザー ID およびパスワードにマップする方法を示す例を示します。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を Microsoft SQL Server リモート・サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR elizabeth SERVER sqlserver
  OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

elizabeth

Microsoft SQL Server リモート・サーバーで定義されたユーザー ID にマップする、ローカルのユーザー ID を指定します。

SERVER *sqlserver*

CREATE SERVER ステートメントで定義した Microsoft SQL Server リモート・サーバーの名前を指定します。

REMOTE_AUTHID '*liz*'

elizabeth をマップする、Microsoft SQL Server リモート・サーバー側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

REMOTE_PASSWORD '*abc123*'

'*liz*' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER sqlserver
  OPTIONS (REMOTE_AUTHID 'liz', REMOTE_PASSWORD 'abc123')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 335 ページの『Microsoft SQL Server データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

Microsoft SQL Server リモート・サーバーへの接続のテスト

Microsoft SQL Server リモート・サーバーへの接続のテストは、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

Microsoft SQL Server リモート・サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、Microsoft SQL Server システム表に対して SQL SELECT ステートメントを発行します。

例:

```
SET PASSTHRU remote_server_name
SELECT count(*) FROM dbo.sysobjects
SET PASSTHRU RESET
```

SQL SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - Microsoft SQL Server リモート・サーバーを調べ、サーバーが開始されていることを確認します。
 - Microsoft SQL Server リモート・サーバーを調べ、入ってくる接続に備えた構成がされていることを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD オプションが Microsoft SQL Server リモート・サーバーとの接続に有効な設定になっているかを確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の ODBC ドライバーを調べ、Microsoft SQL Server リモート・サーバーに接続するように正しくインストールされ構成されていることを確認します。
 - DB2 フェデレーテッド変数の設定を調べ、Microsoft SQL Server リモート・サーバーに合わせた設定になっていることを確認します。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
 - サーバー定義を調べます。必要であれば、それをドロップして再作成します。

この一連のタスクの次のタスクは、『Microsoft SQL Server 表およびビューのニックネームの登録』です。

関連タスク:

- 325 ページの『フェデレーテッド・サーバーへの Microsoft SQL Server データ・ソースの追加』
- 338 ページの『Microsoft SQL Server 表およびビューのニックネームの登録』
- 328 ページの『Microsoft SQL Server 環境変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

Microsoft SQL Server 表およびビューのニックネームの登録

Microsoft SQL Server の表およびビューのニックネームの登録は、フェデレーテッド・サーバーに Microsoft SQL Server データ・ソースを追加するという、より大きな作業の一部を成しています。

定義する Microsoft SQL Server リモート・サーバーそれぞれで、アクセスしたい表またはビューごとにニックネームを登録します。Microsoft SQL Server リモート・サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、CREATE NICKNAME ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録した時に収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れます。オプティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用するため、ニックネームを登録する前に、データ・ソース側で (DB2 の **RUNSTATS** と同等のデータ・ソース・コマンドを使用して) 統計を更新してください。

手順:

ニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME mss_name FOR sqlserver."remote_schema"."remote.table"
```

ニックネームの長さは 128 文字までです。

ニックネームを作成したい Microsoft SQL Server の表またはビューごとに、このステップを繰り返します。

ニックネームを作成すると、DB2 は接続を使用してデータ・ソースのカタログ表を照会します (Microsoft SQL Server では、これらをシステム表と呼びます)。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連資料:

- 「コマンド・リファレンス」の『RUNSTATS コマンド』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 339 ページの『CREATE NICKNAME ステートメント - Microsoft SQL Server ラッパーの例』

CREATE NICKNAME ステートメント - Microsoft SQL Server ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して、アクセスしたい Microsoft SQL Server の表またはビューのニックネームを登録する方法の例を示します。

この例では、ニックネームが割り当てられる Microsoft SQL Server リモート・サーバーのリモート・オブジェクトを指定する方法が示されます。

```
CREATE NICKNAME cust_africa FOR sqlserver.customers.egypt
```

cust_africa

Microsoft SQL Server の表またはビューの固有なニックネーム。

注: ニックネームは、スキーマとニックネームを組み込んだ 2 つの部分からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを作成するユーザーの許可 ID になります。

sqlserver.customers.egypt

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *sqlserver* は、CREATE SERVER ステートメントで Microsoft SQL Server データベース・サーバーに割り当てた名前です。
- *customers* は、表またはビューが属するリモート・スキーマの名前です。
- *egypt* は、アクセスしたいリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Microsoft SQL Server スキーマおよび表の名前を大文字に変換します。

関連タスク:

- 338 ページの『Microsoft SQL Server 表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

Microsoft SQL Server データ・ソースのための構成のチューニングとトラブルシューティング

Microsoft SQL Server データ・ソースに対する構成をセットアップした後、パフォーマンスを改善するために構成を変更したい場合があります。

ODBC トレースの入手

データ・ソースのアクセスに問題がある場合、ODBC トレース情報を入手して問題を分析し、解決することができます。しかしながら、トレースをオンにすると、システム・パフォーマンスに影響が出ます。問題が解決したら、トレースはオフにすべきです。

Windows が稼働するフェデレーテッド・サーバーでは、ODBC トレースが確実に適切に働くようにするため、「ODBC データ ソース アドミニストレータ」が提供するトレース・ツールを使用してください。

UNIX が稼働するフェデレーテッド・サーバーでは、odbc.ini ファイルを変更することによってトレースをオンに設定できます。たとえば、DataDirect ODBC 3.x ドライバーを使用している場合、クライアント・ディレクトリーで odbc.ini ファイルの例を見つけてください。odbc.ini ファイルには、トレース・ファイルに必要な例が含まれています。

```
[ODBC]
Trace=0
TraceFile=/home/user1/trace_dir/filename.xxx
TraceDll==ODBC_driver_directory/odbctrac.so
InstallDir=/opt/odbc
```

トレースを ON にするには、1 行目を Trace=1 に設定します。トレースを OFF にするには、1 行目を Trace=0 に設定します。TraceFile は、このフェデレーテッド・データベース・インスタンスが書き込みアクセス権を持つパスおよびファイル名を指す必要があります。

関連タスク:

- 325 ページの『フェデレーテッド・サーバーへの Microsoft SQL Server データ・ソースの追加』

関連資料:

- 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』

第 17 章 ODBC データ・ソースへのアクセスの構成

この章では、ODBC データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。ODBC データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

ODBC のフェデレーテッド・システムへの追加

フェデレーテッド・サーバーへの ODBC データ・ソースの追加

ODBC データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスしたいデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供しなければなりません。

ODBC データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、DB2 コントロール・センターか DB2 コマンド行を使用することができます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

ODBC API を通してアクセスされるデータ・ソースのことを、ここでは ODBC データ・ソースと呼びます。

必要に応じて、Excel ラッパーを使用する代わりに ODBC ラッパーを使用して Excel データにアクセスできます。Excel データにアクセスするように ODBC ラッパーを構成する特定のステップについては、別のトピックで説明されています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- ODBC ドライバーがフェデレーテッド・サーバー上にインストール・構成されていなければなりません。ODBC ラッパーは、ODBC 3.x をサポートしています。
- システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数の適切なセットアップ。ご使用の ODBC クライアントに必要な変数については、各ベンダーの資料を参照してください。LIBPATH 変数が必要な場合があります。

制約事項:

- ODBC ラッパーを使用して DB2 ファミリーのデータ・ソースにアクセスすることはできません。DB2 ファミリーのデータ・ソースにアクセスするには、DRDA ラッパーを使用してください。
- ODBC ラッパーは、以下の関数やステートメントをサポートしていません。
 - ニックネームでの LOCK TABLE ステートメント
 - ODBC 3.x で推奨されていない機能
 - X/Open または SQL/CLI ドライバー
 - ストアード・プロシージャのニックネーム
 - リモート・セーブポイント・ステートメントを使用した、ステートメント・レベルでの原子性の制約
 - 64 ビット・クライアント
 - WITH HOLD カーソル
- 位置決めされた更新および削除操作をサポートしないデータ・ソースの場合、非 NULL 可能な列に対するユニーク索引がニックネームや該当するリモート表で存在しないと、ニックネームにおいて、位置決めされた UPDATE および DELETE ステートメントや、一部の検索済み UPDATE および DELETE ステートメントは失敗します。こうしたステートメントが失敗した場合は、エラー SQL30090 理由コード 21 が戻されます。
- ODBC ラッパーは、接続ごとのアクティブなステートメントの数を制限するデータ・ソースに対する INSERT、UPDATE、DELETE ステートメントをサポートしません。データ・ソースが接続ごとのアクティブ・ステートメントの数を制限するかどうかを判断するには、そのデータ・ソースのための文書を調べてください。この制限は IBM Red Brick Warehouse にあてはまります。

手順:

ODBC データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. フェデレーテッド・サーバーとフェデレーテッド・データベースを準備します。
2. ラッパーを登録します。
3. サーバー定義を登録します。
4. ユーザー・マッピングを作成します。
5. ODBC データ・ソースとの接続をテストします。
6. ODBC データ・ソースの表およびビューのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』
- 29 ページの『Excel データにアクセスする方法』

関連タスク:

- 353 ページの『ODBC ラッパーを使用した Excel データへのアクセス』
- 57 ページの『フェデレーテッド・データベースの作成』
- 343 ページの『ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備』

- 344 ページの『ODBC ラッパーの登録』
- 346 ページの『ODBC データ・ソースのサーバー定義の登録』
- 348 ページの『ODBC データ・ソースのユーザー・マッピングの作成』
- 350 ページの『ODBC データ・ソース・サーバーへの接続のテスト』
- 351 ページの『ODBC データ・ソース表およびビューのニックネームの登録』

関連資料:

- 28 ページの『正しいラッパーの選択』

ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備

ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備は、ODBC データ・ソースをフェデレーテッド・サーバーに追加するといふ、より大きな作業の一部を成しています。

ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備に必要なステップは、フェデレーテッド・サーバーで稼働しているオペレーティング・システムによって異なります。

注: 使用する ODBC ドライバーとオペレーティング・システムには、固有なライブラリー・パス・ロケーションがあります。

手順:

フェデレーテッド・サーバーの準備は、以下のように行います。

Windows の場合:

フェデレーテッド・サーバー上に、ODBC 3.x ドライバーがインストールされていて構成されているか調べてください。ODBC データ・ソースのノード名が、システム DSN に定義されている必要があります。インストールと構成の手順については、ODBC ドライバーの資料を参照してください。

Microsoft ODBC データ ソース アドミニストレータを使用して DSN を構成した場合は、「コントロール パネル」でこの設定を検査できます。「システム DSN」として ODBC データ・ソースが登録されているか確認してください。そうしないと DB2 が DSN を検出できない可能性があります。

UNIX の場合:

1. ODBC クライアントの構成方法については、ODBC クライアントのベンダーからの資料を参照してください。
2. クライアントが DataDirect ODBC か RedBrick である場合、適切なシンボリック・リンクが作成されていることを確認します。
 - HP-UX の場合、シンボリック・リンクは、`/usr/exe/libodbcinst.sl` から `$ODBC_LIBRARY_PATH/libodbcinst.sl` へのものです。
 - Linux の場合、シンボリック・リンクは、`/usr/lib/libodbcinst.so` から `$DJX_ODBC_LIBRARY_PATH/libodbcinst.so` へのものです。

- Solaris の場合、シンボリック・リンクは、\$HOME/sql1lib/locale から \$DJX_ODBC_LIBRARY_PATH/./locale へのものです。\$HOME は、DB2 インスタンス所有者のホーム・ディレクトリーです。

この一連のタスクの次のタスクは、『ODBC ラッパーの登録』です。

関連タスク:

- 344 ページの『ODBC ラッパーの登録』

ODBC ラッパーの登録

ODBC ラッパーの登録は、フェデレーテッド・サーバーに ODBC データ・ソースを追加するという、より大きな作業の一部を成しています。

ODBC データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER ODBC
```

推奨: ODBC というデフォルトのラッパー名を使用してください。デフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーは、そのラッパーに関連付けられたデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で `odbc_wrapper` という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER odbc_wrapper  
LIBRARY 'libdb2rcodbc.a' OPTIONS (MODULE '/usr/lib/odbc.a')
```

MODULE '`/usr/lib/odbc.a`' は、ODBC Driver Manager を含むライブラリーの絶対パスです。

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『ODBC ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連のタスクの次のタスクは、『ODBC ラッパーのサーバー定義の登録』です。

関連タスク:

- 346 ページの『ODBC データ・ソースのサーバー定義の登録』

関連資料:

- 345 ページの『ODBC ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』
- 345 ページの『CREATE WRAPPER ステートメント - ODBC ラッパーの例』

ODBC ラッパー・ライブラリー・ファイル

次の表は、ODBC ラッパーのためのディレクトリー・パスとライブラリー・ファイル名をリストします。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2rcodbc.a、libdb2rcodbcF.a、および libdb2rcodbcU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 78. ODBC クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2rcodbc.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2rcodbc.sl
Linux	/opt/IBM/db2/V8.1/lib	libdb2rcodbc.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2rcodbc.so
Windows	%DB2PATH%\bin	db2rcodbc.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 344 ページの『ODBC ラッパーの登録』

CREATE WRAPPER ステートメント - ODBC ラッパーの例

このトピックでは、CREATE WRAPPER ステートメントを使用して ODBC データ・ソースのラッパーを登録する方法の例が提供されます。

UNIX システムの場合の例

以下に示すのは、UNIX オペレーティング・システムで CREATE WRAPPER ステートメントを発行してラッパーを登録する方法の例です。

```
CREATE WRAPPER odbc OPTIONS (MODULE '/usr/lib/odbc.so')
```

この例では、*odbc* はフェデレーテッド・データベースに登録するラッパーに割り当てる名前です。MODULE '*usr/lib/odbc.so*' は、ODBC Driver Manager を含むライブラリーの絶対パスです。

UNIX オペレーティング・システムでは MODULE オプションを指定する必要があります。Windows では、MODULE オプションはデフォルトで '*odbc32.dll*' になります。

Windows の場合の例

以下に示すのは、Windows オペレーティング・システムで CREATE WRAPPER ステートメントを発行してラッパーを登録する方法の例です。

```
CREATE WRAPPER odbc LIBRARY 'db2rcodbc.dll'
```

この例では、*odbc* はフェデレーテッド・データベースに登録するラッパーに割り当てる名前です。LIBRARY '*db2rcodbc.dll*' は、ODBC ラッパーのためのライブラリー名です。

関連タスク:

- 344 ページの『ODBC ラッパーの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

ODBC データ・ソースのサーバー定義の登録

ODBC データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに ODBC データ・ソースを追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内で、アクセスしたいそれぞれの ODBC データ・ソース・サーバーを定義する必要があります。

手順:

ODBC データ・ソースのサーバー定義を登録するには、次のようにします。

CREATE SERVER ステートメントを発行します。

例:

```
| CREATE SERVER server_name TYPE data_source_type  
|         VERSION version WRAPPER odbc  
|         OPTIONS (NODE 'node_name')  
|
```

TYPE パラメーターと VERSION パラメーターはオプションです。

NODE は CREATE SERVER ステートメントでオプションとなっていますが、ODBC データ・ソースの場合は必要です。

サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

この一連のタスクの次のタスクは、『ODBC データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 348 ページの『ODBC データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 347 ページの『CREATE SERVER ステートメント - ODBC ラッパーの例』

CREATE SERVER ステートメント - ODBC ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して ODBC ラッパー用サーバーを登録する方法を示す例が提供されます。必要パラメーターを使用してサーバーを登録する方法の完全指定の例、およびその他のサーバー・オプションの例がこのトピックには含まれています。

完全指定の例:

以下に示すのは、CREATE SERVER ステートメントを発行して、ODBC ラッパーのサーバー定義を登録する方法の例です。

```
CREATE SERVER mysql_server TYPE mysql
  VERSION 4.0 WRAPPER odbc
  OPTIONS (NODE 'odbc_node', DBNAME 'venice')
```

mysql_server

ODBC データ・ソース・サーバーに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE *mysql*

アクセスを構成しているデータ・ソースのタイプを指定します。このパラメーターはオプションです。

VERSION 4.0

アクセスしたい ODBC データ・ソースのバージョン。このパラメーターはオプションです。

WRAPPER *odbc*

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE '*mysql_node*'

DSN を定義した時に ODBC データ・ソースに割り当てたノード名 (システム DSN 名)。この値は大文字小文字の区別があります。Windows では、この値は「ODBC データ管理 (ODBC Data Administration)」ウィンドウにあるシステム DSN 名でなければなりません。UNIX では、使用する値の情報について、ODBC クライアント・ベンダーの資料を参照してください。

NODE は CREATE SERVER ステートメントでオプションとなっていますが、ODBC データ・ソースの場合は必要です。

DBNAME 'venice'

アクセスしたいデータベースの名前。この値は大文字小文字の区別がありません。

サーバー・オプションの例:

以下に示すのは、DB2_TABLE_QUOTE_CHAR、DB2_ID_QUOTE_CHAR、および DB2_AUTHID_QUOTE_CHAR サーバー・オプションの使用法を示す例です。

一部の ODBC データ・ソースは (例: MySQL)、SQL ステートメントで表名や列名を囲んでいる引用符を処理できません。これらのデータ・ソースにアクセスするには、CREATE SERVER ステートメントに以下のサーバー・オプションを含める必要があります。

- DB2_TABLE_QUOTE_CHAR '``'
- DB2_ID_QUOTE_CHAR '``'
- DB2_AUTHID_QUOTE_CHAR '``'

、文字はスキーマ名、表名、列名などの ID の区切り文字です。

例:

```
CREATE SERVER mysql_server TYPE mysql
      VERSION 4.0 WRAPPER odbc
      OPTIONS (NODE 'mysql_node', DB2_TABLE_QUOTE_CHAR '``',
              DB2_ID_QUOTE_CHAR '``' DB2_AUTHID_QUOTE_CHAR '``')
```

関連タスク:

- 346 ページの『ODBC データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

ODBC データ・ソースのユーザー・マッピングの作成

ODBC データ・ソースのユーザー・マッピングの作成は、ODBC データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

ODBC データ・ソースにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ローカル・ユーザー ID を ODBC データ・ソースのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR userid SERVER server_name
  OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

この一連のタスクの次のタスクは、『ODBC データ・ソース・サーバーへの接続のテスト』です。

関連タスク:

- 350 ページの『ODBC データ・ソース・サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 349 ページの『CREATE USER MAPPING ステートメント - ODBC ラッパーの例』

CREATE USER MAPPING ステートメント - ODBC ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、ローカルのユーザー ID を ODBC データ・ソースのユーザー ID およびパスワードにマップする方法を示す例が提供されます。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下に示すのは、ローカル・ユーザー ID を ODBC データ・ソースのユーザー ID およびパスワードにマップする方法の例です。

```
CREATE USER MAPPING FOR arturo SERVER server_name
  OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

arturo ODBC データ・ソースで定義されたユーザー ID にマップするローカルのユーザー ID を指定します。

server_name

CREATE SERVER ステートメントで定義した ODBC データ・ソースの名前を指定します。

'art' *arturo* をマップする ODBC データ・ソース側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

'red4blue'

'art' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

以下に、特殊レジスター `USER` を含む `CREATE USER MAPPING` ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER server_name
    OPTIONS (REMOTE_AUTHID 'art', REMOTE_PASSWORD 'red4blue')
```

DB2 特殊レジスター `USER` を使用して、`CREATE USER MAPPING` ステートメントを実行する人の許可 ID を、`REMOTE_AUTHID` ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 348 ページの『ODBC データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

ODBC データ・ソース・サーバーへの接続のテスト

ODBC データ・ソース・サーバーへの接続のテストは、ODBC データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

ODBC データ・ソース・サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

前提条件:

使用するデータ・ソースは、パススルー・セッションをサポートしている必要があります。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、ODBC データ・ソース・システム表に対して SQL `SELECT` ステートメントを発行します。

例:

```
SET PASSTHRU server_name
SELECT COUNT(*) FROM schema_name.table_name
SET PASSTHRU RESET
```

server_name は、`CREATE SERVER` ステートメントで定義した ODBC データ・ソースの名前です。

schema_name は、リモート ODBC データ・ソースでのスキーマ名です。ODBC データ・ソースがスキーマをサポートしていない場合、ステートメントからスキーマを省略してください。

table_name は、リモート ODBC データ・ソースでの表の名前です。

SQL `SELECT` ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - データ・ソースが使用可能であることを確認します。
 - 必要であれば、データ・ソース・サーバーを調べ、入ってくる接続に備えて構成されているかどうかを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD のオプションが ODBC データ・ソースとの接続に有効な設定になっているか確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の ODBC ドライバーを調べ、ODBC データ・ソース・サーバーに接続するように正しくインストールされて構成されていることを確認します。Windows オペレーティング・システムでは、ドライバーをチェックするときに「ODBC データ ソース アドミニストレータ」ツールを使用します。UNIX オペレーティング・システムの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。
 - サーバー定義を調べます。必要であれば、それをドロップして再作成します。

この一連のタスクの次のタスクは、『ODBC データ・ソース表およびビューのニックネームの登録』です。

関連タスク:

- 341 ページの『フェデレーテッド・サーバーへの ODBC データ・ソースの追加』
- 343 ページの『ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備』
- 351 ページの『ODBC データ・ソース表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

ODBC データ・ソース表およびビューのニックネームの登録

ODBC データ・ソースの表およびビューのニックネームの登録は、フェデレーテッド・サーバーに ODBC データ・ソースを追加するという、より大きな作業の一部を成しています。

定義するそれぞれの ODBC データ・ソース・サーバーで、アクセスしたい各表またはビューごとにニックネームを登録します。ODBC データ・ソースを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

たとえば、Microsoft SQL Server の *italy* と呼ばれる表 (スキーマ名は *customers*) に、ニックネーム *cust_europe* を定義するとします。こうすると、フェデレーテッド・サーバーから `SELECT * FROM cust_europe` という SQL ステートメントを使用することができます。ただし、`SELECT * FROM server_name."customers"."italy"` というステートメントは使用できません。

ODBC データ・ソースの表とビューのニックネームの登録に加えて、リモート・システムの表のニックネームの登録もできます。

ODBC データ・ソースがスキーマをサポートしていない場合、**CREATE NICKNAME** ステートメントからスキーマを省略してください。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、**CREATE NICKNAME** ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録した時に収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オブティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れられます。オブティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用するため、ニックネームを登録する前に、データ・ソース側で (DB2 の **RUNSTATS** と同等のデータ・ソース・コマンドを使用して) 統計を更新してください。

制約事項:

索引がある ODBC データ・ソースにニックネームを作成する場合、ODBC ラッパーは索引情報をフェデレーテッド・データベースのシステム・カタログに記録しません。SPECIFICATION ONLY 文節を指定した **CREATE INDEX** ステートメントを使用することによって、表に対する索引の指定を作成しなければなりません。

手順:

ニックネームを登録するには、**CREATE NICKNAME** ステートメントを発行します。

例:

```
CREATE NICKNAME odbc_name FOR server_name."remote_schema"."remote.table"
```

ニックネームの長さは 128 文字までです。

ニックネームを作成したい ODBC の表またはビューごとに、このステップを繰り返します。

ニックネームを作成すると、DB2 は接続を使用してデータ・ソースのカタログ表を照会します。この照会は、ニックネームを使用して ODBC データ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連タスク:

- 「フェデレーテッド・システム・ガイド」の『データ・ソース・オブジェクト用の索引の指定の作成』

関連資料:

- 「コマンド・リファレンス」の『RUNSTATS コマンド』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 353 ページの『CREATE NICKNAME ステートメント - ODBC ラッパーの例』

CREATE NICKNAME ステートメント - ODBC ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して、アクセスしたい ODBC データ・ソースの表またはビューのニックネームを登録する方法の例が提供されます。

この例では、ニックネームが割り当てられる ODBC データ・ソース・リモート・サーバーのリモート・オブジェクトを指定する方法が示されます。

```
CREATE NICKNAME cust_europe FOR server_name."customers"."italy"
```

cust_europe

表またはビューの固有なニックネーム。ニックネームはスキーマ内で固有でなければなりません。

注: ニックネームは、スキーマとニックネームを組み込んだ 2 つの部分からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

server_name."customers"."italy"

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *server_name* は、CREATE SERVER ステートメントで ODBC データベース・サーバーに割り当てた名前です。
- *customers* は、表またはビューが属するリモート・スキーマの名前です。ODBC データ・ソースがスキーマをサポートしていない場合、CREATE NICKNAME ステートメントからスキーマを省略してください。
- *italy* は、アクセスしたいリモートの表またはビューの名前です。

ODBC データ・ソース・オブジェクトでは、大文字と小文字が区別される場合があります。リモート・スキーマ名とリモート表名の両方を引用符で囲んでください。そうでない場合、DB2 はこれらの名前を大文字に変換します。

関連タスク:

- 351 ページの『ODBC データ・ソース表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

ODBC ラッパーを使用した Excel データへのアクセス

Excel ODBC ドライバーを使用すると、ODBC ラッパーで Microsoft Excel ワークブックにアクセスできます。Excel アプリケーションをフェデレーテッド・サーバー上にインストールする必要はありません。Excel ODBC ドライバーは、自動的に Microsoft Windows と共にインストールされます。

ODBC ラッパーと Excel ODBC ドライバーを使用すると、ワークブック中のすべてのワークシート中のデータにアクセスできます。Excel ODBC ドライバーは、ワークブックをデータベースと解釈し、ワークブック中の個々のワークシートを表と解釈します。

ワークブックを作成したバージョンの Excel アプリケーションがサポートされなくなっても、Excel ODBC ドライバーは旧バージョンの Excel ワークブックをサポートしています。たとえば、Microsoft 社は Excel バージョン 4.0 で作成されたワークシートをサポートしなくなりましたが、このドライバーはこのバージョンで作成した Excel ワークシートをサポートしています。

前提条件:

Excel ODBC ドライバーがフェデレーテッド・サーバー上になければなりません。

フェデレーテッド・サーバーは Excel ワークブック中のワークシートを開き読み取って、データを取り出せなければなりません。したがって、Excel ワークブックは DB2 フェデレーテッド・サーバーと同じコンピューター上か、またはアクセス可能なマップされたネットワーク・ドライブ上になければなりません。

制約事項:

- ワークブックがユーザーまたはアプリケーションによって読み取り/書き込みモードですでに開かれている場合、ODBC ラッパーはワークシートにアクセスできません。しかし、ユーザーまたはアプリケーションがワークブックを開く前に ODBC ラッパーがワークブックを開いた場合は、ユーザーまたはアプリケーションは読み取り専用モードでワークブックを開くことができます。
- Excel ODBC ドライバーは、最初の実行可能行にワークシート列のラベルが含まれていることを予期しています。ワークシートにラベルがない場合は、ワークシート中に列ラベルの行を挿入しなければなりません。
- Excel ODBC ドライバーは Windows オペレーティング・システムでだけ入手可能なため、ODBC ラッパーを使用してアクセスできるのは、Windows オペレーティング・システムが稼働しているフェデレーテッド・サーバー上の Excel データだけです。
- Excel ワークシート上で挿入操作と更新操作を実行できますが、削除操作は実行できません。Excel ODBC ドライバーは、削除操作をサポートしていません。ワークシートからデータを削除するには、Excel でワークシートを開いて変更を加えなければなりません。

手順:

ODBC ラッパーを使用して Excel ワークシートにアクセスするには、以下のようになります。

1. アクセスしたい Excel ワークブックが、フェデレーテッド・サーバー上にあるか、アクセス可能なマップされたネットワーク・ドライブにあるか確認します。
2. Excel データにアクセスする方法
 - 必要に応じて、Excel ワークシート中のデータのレイアウトに変更を加えて、Excel ODBC ドライバーの要件を満たすようにします。
 - 必要に応じて、アクセスしたい名前付き範囲を作成します。
3. アクセスしたいワークブックのシステム DSN を作成します。システム DSN の構成には、「ODBC データ ソース アドミニストレータ」を使用できます。システム DSN の作成時に指定した名前が、CREATE SERVER ステートメント中の NODE オプションの値として割り当てられます。
4. CREATE WRAPPER ステートメントを発行します。

例:

```
CREATE WRAPPER odbc
```

5. フェデレーテッド・データベース・システム・カタログ中にサーバー・オブジェクトを登録して、ワークブックの場所を指定します。 ODBC ラッパーの場合は、DSN ごとにサーバー・オブジェクトが必要です。 Excel ODBC ドライバーの使用時に DSN がワークブックと関連付けられます。

ワークブックの場所を指定するには、CREATE SERVER ステートメントを発行し、NODE オプションのシステム DSN として DSN を使用します。

例:

```
CREATE SERVER compounds_workbook WRAPPER odbc
  OPTIONS (NODE 'compounds_workbook_dsn', PASSWORD 'n')
```

NODE *compounds_workbook_dsn* は、作成したシステム DSN です。 ODBC ラッパーが Excel ワークシートにアクセスするには、NODE オプションと PASSWORD オプションが必要です。

アクセスしようとしているワークブックごとに、このステップを繰り返します。

6. CREATE NICKNAME ステートメントを発行して、アクセスしたいワークシートのニックネームを作成します。構文は以下のとおりです。

```
CREATE NICKNAME nickname FOR server_name.remote_table
```

- 名前付き範囲を作成してデータにアクセスする場合は、CREATE NICKNAME ステートメントの *remote_table* の部分に範囲の名前を指定します。

たとえば、範囲の名前が *testing* の場合は、CREATE NICKNAME ステートメントは以下のようになります。

```
CREATE NICKNAME compounds_nickname FOR compounds_workbook.testing
```

- 範囲ではなくワークシート全体のデータにアクセスするには、ワークシートの名前の後に \$ 記号を指定します。

たとえば、ワークシートの名前が *Sheet1* の場合は、CREATE NICKNAME ステートメントは以下のようになります。

```
CREATE NICKNAME compounds_nick FOR compounds_workbook.Sheet1$
```

アクセスしたいワークシートまたは名前付き範囲ごとに、このステップを繰り返します。

デフォルトのデータ・タイプ・マッピングの変更

ODBC ラッパーを使用する場合は、サポートされているデータ・タイプが Excel ODBC ドライバーによって判別されます。 Excel ODBC ドライバーは、Excel データ・タイプを ODBC データ・タイプにマップします。次に ODBC ラッパーは ODBC データ・タイプを DB2 データ・タイプにマップします。列ごとの DB2 データ・タイプは、フェデレーテッド・データベース・カタログ表に保管されます。これらのデータ・タイプのことを、ローカル・データ・タイプ と呼びます。次の表は、デフォルトのデータ・タイプ・マッピングのリストです。

表 79. ODBC ラッパー使用時の Excel と DB2 の間のデフォルト・データ・タイプ・マッピング

Excel データ・タイプ	ODBC データ・タイプ	DB2 データ・タイプ
CURRENCY	SQL_NUMERIC	DECIMAL/DOUBLE
DATETIME	SQL_TIMESTAMP	TIMESTAMP
LOGICAL	SQL_BIT	SMALLINT
NUMBER	SQL_DOUBLE	DOUBLE
TEXT	SQL_VARCHAR	VARCHAR

しかし、ODBC ラッパーは多種類のデータ・タイプをサポートしています。デフォルトのデータ・タイプを他の DB2 データ・タイプにマップできます。ローカル・データ・タイプに変更を加えるには、ALTER NICKNAME ステートメントを使用します。ODBC ラッパーで使用できるデータ・タイプのリストは、包括的なもので、LOB や他の 2 バイトのデータ・タイプなどのデータ・タイプが含まれます。

重要: ローカル・データ・タイプをオリジナルのマッピング以外のデータ・タイプに変更すると、データ・タイプの不一致が生じる可能性があります。

ALTER SERVER ステートメント - ODBC ラッパーで Excel データにアクセスする例

Excel ODBC ドライバーにより、ODBC ラッパーを使用して述部や集約関数をデータ・ソースにプッシュダウンして処理できます。このドライバーは、ワークシート間の結合もサポートしています。データ・ソースのプッシュダウン処理を行うと、照会でワークシートを参照する際にデータ・ソースの行が一部しか DB2 に戻されないの、パフォーマンスを改善できます。

サーバー *compounds_workbook* を定義して、アクセスしたいデータを含むワークブックを識別したとします。このワークブックの場合、フェデレーテッド・サーバーが Excel ODBC ドライバーに操作をプッシュダウンして処理できるよう指定したいとします。ALTER SERVER ステートメントは、次のようになります。

```
ALTER SERVER compounds_workbook
  OPTIONS (ADD PUSHDOWN 'Y')
```

プッシュダウン機能の利点を活用するには、CREATE SERVER ステートメントまたは ALTER SERVER ステートメントを使用して、以下の ODBC サーバー・オプションを設定する必要があります。

- *PUSHDOWN* 'Y'
- *DB2_GROUP_BY* 'Y'
- *DB2_COLFUNC* 'Y'

関連概念:

- 29 ページの『Excel データにアクセスする方法』

関連タスク:

- 343 ページの『ODBC を通してデータ・ソースにアクセスするためのフェデレーテッド・サーバーの準備』

関連資料:

ODBC データ・ソースのための構成のチューニングとトラブルシューティング

ODBC データ・ソースに対する構成をセットアップした後、パフォーマンスを改善するために構成を変更したい場合があります。

ODBC トレースの入手

データ・ソースのアクセスに問題がある場合、ODBC トレース情報を入手して問題を分析し、解決することができます。しかしながら、トレースをオンにすると、システム・パフォーマンスに影響が出ます。問題が解決したら、トレースはオフにすべきです。

Windows が稼働するフェデレーテッド・サーバーでは、「ODBC データ ソース アドミニストレータ」ウィンドウの「トレース」をクリックします。

UNIX が稼働するフェデレーテッド・サーバーの場合は、ODBC クライアントのベンダーが提供している資料を参照してください。

関連タスク:

- ・ 341 ページの『フェデレーテッド・サーバーへの ODBC データ・ソースの追加』

関連資料:

- ・ 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』

第 18 章 OLE DB データ・ソースへのアクセスの構成

この章では、OLE DB データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。OLE DB データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要のある作業のリスト、および必要な SQL ステートメントの例を示します。

フェデレーテッド・サーバーへの OLE DB データ・ソースの追加

Microsoft OLE DB は、さまざまな情報源に保管されたデータに、統一された形式でアプリケーションがアクセスできるようにする OLE/COM インターフェースのセットです。OLE DB コンポーネントの DBMS アーキテクチャーでは、OLE DB コンシューマーと OLE DB プロバイダーを定義しています。OLE DB コンシューマーとは、OLE DB インターフェースを使用するシステムまたはアプリケーションです。OLE DB プロバイダーとは、OLE DB インターフェースを提供するコンポーネントです。

OLE DB ラッパーにより、Microsoft OLE DB 2.0 以上に準拠する OLE DB プロバイダーにアクセスできるようになります。

OLE DB ラッパーは、表関数の作成に使用します。データ・ソースの表やビューにニックネームを作成する場合には、ラッパーは使用できません。

フェデレーテッド・サーバーが OLE DB データ・ソースにアクセスするように構成するには、OLE DB プロバイダーに関する情報をフェデレーテッド・サーバーに提供しなければなりません。

OLE DB データ・ソースへのアクセスは、DB2 コマンド・センターまたは DB2 コマンド行を通して構成できます。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

制約事項:

OLE DB ラッパーは、ユーザー定義の OLE DB 外部表関数の登録を援助するためにだけ使用されます。他のラッパーとは異なり、OLE DB ラッパーは、データ・ソースに保管されているデータへのアクセスにニックネームを使用しません。

手順:

フェデレーテッド・サーバーに OLE DB データ・ソースを追加するには、以下のようになります。

1. ラッパーを登録します。
2. サーバー定義を登録します。
3. ユーザー・マッピングを作成します。

OLE DB データ・ソースへのアクセスを構成した後、CREATE FUNCTION ステートメントを使用して、ユーザー定義の OLE DB 外部表関数をフェデレーテッド・データベースに登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 360 ページの『OLE DB ラッパーの登録』

OLE DB ラッパーの登録

OLE DB ラッパーの登録は、フェデレーテッド・サーバーに OLE DB データ・ソースを追加するという、より大きな作業の一部を成しています。

OLE DB データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER OLEDB
```

推奨: OLEDB というデフォルトのラッパー名を使用してください。デフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーは、そのラッパーに関連付けられたデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、Windows オペレーティング・システムを使用するフェデレーテッド・サーバー上で oledb_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER oledb_wrapper LIBRARY 'db2oledb.dll'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『OLE DB ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連のタスクの次のタスクは、『OLE DB ラッパーのサーバー定義の登録』です。

関連タスク:

- 361 ページの『OLE DB データ・ソースのサーバー定義の登録』

関連資料:

- 361 ページの『OLE DB ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

OLE DB ラッパー・ライブラリー・ファイル

次の表は、OLE DB ラッパーのためのディレクトリー・パスとライブラリー・ファイル名をリストします。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスにライブラリー・ファイルが追加されます。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 80. OLE DB クライアント・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
Windows	%DB2PATH%\%bin	db2oledb.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 360 ページの『OLE DB ラッパーの登録』

OLE DB データ・ソースのサーバー定義の登録

OLE DB データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに OLE DB データ・ソースを追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスしたいそれぞれの OLE DB データ・ソース・サーバーを定義する必要があります。

手順:

OLE DB データ・ソースのサーバー定義を登録するには、次のようにします。

CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name WRAPPER OLEDB
    OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
                        Data Source=c:\msdasdk\bin\oledb\%nwind.mdb')
```

この一連のタスクの次のタスクは、『OLE DB データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 363 ページの『OLE DB データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE SERVER ステートメント』
- 362 ページの『CREATE SERVER ステートメント - OLE DB ラッパーの例』

CREATE SERVER ステートメント - OLE DB ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して OLE DB ラッパー用サーバーを登録する方法を示す例が提供されます。

以下に、CREATE SERVER ステートメントの例を示します。

```
CREATE SERVER Nwind WRAPPER OLEDB
    OPTIONS (CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0;
                        Data Source=c:\msdasdk\bin\oledb\%nwind.mdb',
            COLLATING_SEQUENCE 'Y')
```

Nwind OLE DB データ・ソースに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

WRAPPER OLEDB

CREATE WRAPPER ステートメントで指定したラッパー名。

CONNECTSTRING 'Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\msdasdk\bin\oledb\%nwind.mdb'

データ・ソースへの接続に必要な初期化プロパティを指定します。

このストリングには、一連のキーワードと値の対をセミコロンで区切ったものを入れます。等号 (=) は、各キーワードとその値を区切ります。キーワードは、OLE DB 初期化プロパティ (プロパティ・セット DBPROPSET_DBINT) の記述、またはプロバイダー特有のキーワードです。

CONNECTSTRING オプションの完全な構文とセマンティクスについては、『Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK』(Microsoft Press、1998) を参照してください。

COLLATING_SEQUENCE 'Y'

データ・ソースが DB2 (UNIX および Windows 照合シーケンス) と同じ照合シーケンスを使用するかどうかを指定します。

有効な値は、'Y' (データ・ソースが UNIX および Windows 照合シーケンス用の DB2 を使用する) および 'N' (データ・ソースが UNIX および Windows 照合シーケンス用の DB2 とは異なる照合シーケンスを使用する) です。デフォルト値は 'N' です。

関連タスク:

- 361 ページの『OLE DB データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

OLE DB データ・ソースのユーザー・マッピングの作成

OLE データ・ソースのユーザー・マッピングの作成は、OLE データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

OLE データ・ソースにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ローカル・ユーザー ID を OLE データ・ソースのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR userid SERVER server_name
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

| OLE DB データ・ソースのパスワードかフェデレーテッド・サーバーのパスワード
| のいずれかが 8 文字未満の場合、OLE DB データ・ソースにアクセスする SQL ス
| テートメントは失敗します。次のエラー・メッセージが表示されます。

```
| SQL30082N 接続確立の試行は、セキュリティー上の理由 "15"  
| ("PROCESSING FAILURE") により失敗しました。 SQLSTATE=08001
```

| この問題を避けるには、OLE DB データ・ソースのパスワードかフェデレーテッ
| ド・サーバーのパスワードを 8 文字以上に変更してください。

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 364 ページの『CREATE USER MAPPING ステートメント - OLE DB ラッパーの例』

CREATE USER MAPPING ステートメント - OLE DB ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、ローカルのユーザー ID を OLE データ・ソースのユーザー ID およびパスワードにマップする方法を示す例が提供されます。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下に示すのは、ローカル・ユーザー ID を OLE データ・ソースのユーザー ID およびパスワードにマップする方法の例です。

```
CREATE USER MAPPING FOR laura SERVER Nwind
  OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders')
```

laura OLE DB データ・ソースで定義されたユーザー ID と対応付けるローカルのユーザー ID。

SERVER *Nwind*

CREATE SERVER ステートメントで定義した OLE DB サーバーの名前。

REMOTE_AUTHID '*lulu*'

laura と対応付ける OLE DB サーバー側のユーザー ID。この値は大文字小文字の区別があります。

REMOTE_PASSWORD '*raiders*'

'*lulu*' に関連するパスワード。この値は大文字小文字の区別があります。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER Nwind
  OPTIONS (REMOTE_AUTHID 'lulu', REMOTE_PASSWORD 'raiders')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 363 ページの『OLE DB データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』

第 19 章 Oracle データ・ソースへのアクセスの構成

この章では、Oracle データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Oracle データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

フェデレーテッド・システムへの Oracle の追加

フェデレーテッド・サーバーへの Oracle データ・ソースの追加

Oracle データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスしたいデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに提供しなければなりません。

Oracle データ・ソースをアクセスするようにフェデレーテッド・サーバーを構成するには、DB2 コントロール・センターか DB2 コマンド行を使用することができます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- フェデレーテッド・サーバーにインストール・構成された Oracle クライアント・ソフトウェア。
- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

Oracle データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. Oracle クライアント構成ファイルをセットアップしてテストします。
2. Oracle 環境変数をセットアップします。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Oracle サーバーとの接続をテストします。
7. Oracle の表、ビュー、およびシノニムのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 370 ページの『Oracle クライアント構成ファイルのセットアップおよびテスト』
- 371 ページの『Oracle ラッパーの登録』
- 373 ページの『Oracle データ・ソースのサーバー定義の登録』
- 375 ページの『Oracle データ・ソースのユーザー・マッピングの作成』
- 377 ページの『Oracle サーバーへの接続のテスト』
- 378 ページの『Oracle 表およびビューのニックネームの登録』
- 379 ページの『Oracle データ・ソースのための構成のチューニングとトラブルシューティング』
- 366 ページの『Oracle 環境変数の設定』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』

関連資料:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Oracle 環境変数の設定

Oracle 環境変数の設定は、フェデレーテッド・サーバーに Oracle を追加するという、より大きな作業の一部を成しています。

DB2 Information Integrator のインストール時に、インストール処理は db2dj.ini ファイルに Oracle 環境変数を設定しようとします。

以下の場合には、環境変数は db2dj.ini ファイルに設定されません。

- DB2 フェデレーテッド・サーバーのセットアップ後に Oracle クライアント・ソフトウェアをインストール。
- Oracle クライアント・ソフトウェアがインストールされていない。

Oracle 用に有効な環境変数は以下のとおりです。

- ORACLE_HOME
- ORACLE_BASE (オプション)
- ORA_NLS (オプション)
- TNS_ADMIN (オプション)
- NLS_LANG (オプション)

オプション環境変数は手動で設定する必要があります。

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。『db2dj.ini ファイルの制約事項』

手順:

必要な環境変数を自動的に設定するには、次のようにします。

1. クライアント・ソフトウェアが DB2 フェデレーテッド・サーバーにインストールされていない場合、インストールして構成します。
2. 必要な環境変数を設定します。DB2 Information Integrator のインストールを再実行することによって、環境変数を自動的に設定できます。ランチパッドから、「製品のインストール」をクリックし、ウィザードの指示に従います。

Oracle 環境変数を手動で設定するには、以下のようになります。

1. db2dj.ini ファイルを編集します。
 - Windows が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib¥cfg` ディレクトリーにあります。
 - UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib/cfg` ディレクトリーにあります。

db2dj.ini ファイルには、フェデレーテッド・サーバー上にインストールされている Oracle クライアント・ソフトウェアの構成情報が入っています。このファイルがない場合は、任意のテキスト・エディターを使用して、db2dj.ini という名前の新規ファイルを作成することができます。db2dj.ini ファイルには、変数に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。

2. 必要に応じて以下の環境変数を設定します。

ORACLE_HOME

ORACLE_HOME 環境変数に、Oracle クライアント・ソフトウェアがインストールされているディレクトリー・パスを設定します。この変数 `ORACLE_HOME=oracle_home_directory` には、完全修飾パスを指定します。たとえば、Oracle ホーム・ディレクトリーが `¥usr¥oracle¥8.1.7` である場合、db2dj.ini の項目は `ORACLE_HOME=¥usr¥oracle¥8.1.7` になります。

注: フェデレーテッド・インスタンスの個々のユーザーが ORACLE_HOME 環境変数を設定していても、フェデレーテッド・インスタンスはその設定を使用しません。フェデレーテッド・インスタンスは、db2dj.ini ファイルに設定された ORACLE_HOME の値だけを使用します。

ORACLE_BASE

ORACLE_BASE は、Oracle クライアント・ディレクトリー・ツリーのルートを表します。Oracle クライアント・ソフトウェアをインストール

する際に ORACLE_BASE 変数を設定したら、フェデレーテッド・サーバーに ORACLE_BASE 環境変数を設定します。例:

```
ORACLE_BASE=oracle_root_directory
```

ORA_NLS*

Oracle の複数のバージョンを使用している場合、以下を確認します。

- 適切な ORA_NLS 変数が設定されている。
- 使用するバージョンに対応する NLS データ・ファイルが使用可能である。

ORA_NLS* 環境変数で指定されるディレクトリーに、場所に固有のデータが保存されます。Oracle の新しいバージョンごとに、別の ORA_NLS データ・ディレクトリーがあります。

表 8I. バージョンごとのロケール固有データの場所を指定する Oracle ORA_NLS 変数

Oracle バージョン	環境変数
7.2	ORA_NLS
7.3	ORA_NLS32
8.x, 9.x	ORA_NLS33
10g	ORA_NLS10

たとえば、Oracle 8.1 データ・ソースにアクセスする UNIX フェデレーテッド・サーバーの場合、次のように ORA_NLS33 環境変数を設定します。

```
ORA_NLS33=oracle_home_directory/ocommon/nls/admin/<data>
```

TNS_ADMIN

- Windows が稼働するフェデレーテッド・サーバーの場合、Oracle クライアントは tnsnames.ora ファイルを %ORACLE_HOME%\NETWORK\ADMIN ディレクトリーで探します。ここで %ORACLE_HOME% は db2dj.ini ファイルで定義されます。tnsnames.ora ファイルが %ORACLE_HOME%\NETWORK\ADMIN ディレクトリーにない場合、フェデレーテッド・サーバー上に TNS_ADMIN 環境変数を設定する必要があります。
- AIX、Linux、および HP-UX が稼働するフェデレーテッド・サーバーの場合、クライアントは tnsnames.ora ファイルを /etc ディレクトリーで探します。tnsnames.ora ファイルが /etc ディレクトリーに見つからない場合、クライアントは \$ORACLE_HOME/network/admin ディレクトリーでこのファイルを探します。ここで、\$ORACLE_HOME は db2dj.ini ファイルに定義されています。クライアントが tnsnames.ora ファイルを見つけられない場合、TNS_ADMIN 環境変数を設定する必要があります。この変数は、db2dj.ini ファイルで、tnsnames.ora ファイルが見つかるパスを設定します。

たとえば、tnsnames.ora ファイルが /home/oracle ディレクトリーにある場合、次のようにこの変数を設定します。

```
TNS_ADMIN=/home/oracle
```

- Solaris が稼働するフェデレーテッド・サーバーの場合、クライアントは `tnsnames.ora` ファイルを `/var/opt/oracle` ディレクトリーで探します。`tnsnames.ora` ファイルが `/var/opt/oracle` ディレクトリーに見つからない場合、クライアントは `$ORACLE_HOME/network/admin` ディレクトリーでこのファイルを探します。ここで、`$ORACLE_HOME` は `db2dj.ini` で定義されています。クライアントが `tnsnames.ora` ファイルを見つけない場合、`TNS_ADMIN` 環境変数を設定する必要があります。この変数は、`db2dj.ini` ファイルで、`tnsnames.ora` ファイルが見つかるパスを設定します。

たとえば、`tnsnames.ora` ファイルが `/home/oracle` ディレクトリーにある場合、次のようにこの変数を設定します。

```
TNS_ADMIN=/home/oracle
```

3. UNIX では、DB2 インスタンスの `.profile` ファイルを Oracle 環境変数を使用して更新します。これは次のコマンドを実行して行うことができます。

```
export ORACLE_HOME=oracle_home_directory
export PATH=$ORACLE_HOME/bin:$PATH
```

ここで、`oracle_home_directory` は Oracle クライアント・ソフトウェアがインストールされているディレクトリーです。

4. UNIX の場合、次のように入力して DB2 インスタンス `.profile` を実行します。

```
.$HOME/.profile
```

5. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。DB2 インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop
db2start
```

Oracle コード・ページ変換をセットアップする:

Oracle コード・ページ変換のために、オプションの環境変数 `NLS_LANG` を設定することができます。

Oracle ラッパーが Oracle データ・ソースに接続するたびに、ラッパーはその接続のためにどのコード・ページを使用するかを決定します。フェデレーテッド・サーバー上の `db2dj.ini` ファイルで `NLS_LANG` が設定されている場合、ラッパーは `db2dj.ini` ファイルの値を使用します。`db2dj.ini` ファイルには、フェデレーテッド・サーバー上にインストールされている Oracle クライアント・ソフトウェアの構成情報が入っています。

フェデレーテッド・サーバー上で `NLS_LANG` 変数が設定されていない場合、ラッパーはフェデレーテッド・データベースのテリトリーとコード・ページを判断します。ラッパーは `NLS_LANG` を一致する Oracle ロケールのうち最も近いものに設定します。近く一致するロケールがない場合、`NLS_LANG` は `American_America.US7ASCII` に設定されます。

中国語コード・ページ `GB 18030` を使用するデータを含むデータ・ソースにアクセスする場合、フェデレーテッド・データベースは UTF-8 コード・ページを使用する

必要があります。 Oracle データ・ソースの場合、Oracle ラッパーは Oracle NLS_LANG 環境変数を次のように設定します。

```
NLS_LANG=Simplified Chinese_China.UTF8
```

Oracle 9i クライアントを使用している場合、Oracle 9i クライアントが GB 18030 データを正しく Unicode に変換できるように、db2dj.ini ファイルの NLS_LANG の設定を Simplified Chinese_China.AL32UTF8 に変更してください。

例:

```
NLS_LANG=Simplified Chinese_China.AL32UTF8
```

有効なロケールのリストは、Oracle ソフトウェアに付いてくる資料を参照してください。

関連タスク:

- 378 ページの『Oracle 表およびビューのニックネームの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』
- 66 ページの『db2dj.ini ファイルの制約事項』

Oracle クライアント構成ファイルのセットアップおよびテスト

Oracle クライアント構成ファイルのセットアップとテストは、Oracle データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

クライアント構成ファイルは、フェデレーテッド・サーバーにインストールされているクライアント・ライブラリーを使用して、Oracle データベースに接続するために使用されます。このファイルは、各 Oracle データベース・サーバーのロケーション、およびそのデータベース・サーバーへの接続のタイプ (プロトコル) を指定します。Oracle クライアント構成ファイルのデフォルト名は tnsnames.ora です。

手順:

Oracle クライアント構成ファイルをセットアップおよびテストするには、次のようになります。

1. Oracle クライアント・ソフトウェアに付属してくる Oracle NET8/NET 構成ユーティリティを使用します。

このユーティリティの使用法の詳細は、Oracle のインストール資料を参照してください。tnsnames.ora ファイル内で、SID (または SERVICE_NAME) は Oracle インスタンスの名前であり、HOST は Oracle サーバーが存在するホスト名です。

tnsnames.ora ファイルが作成されるディレクトリーは、フェデレーテッド・サーバーで稼働しているオペレーティング・システムにより異なります。

- UNIX オペレーティング・システムの場合、このファイルのデフォルトのパスと名前は \$ORACLE_HOME/network/admin になります。
- Windows オペレーティング・システムの場合、このファイルのデフォルトのパスおよび名前は %ORACLE_HOME%\network\admin になります。

2. tnsnames.ora ファイルをデフォルトの検索パス以外のパスに置く場合は、TNS_ADMIN 環境変数を設定してファイルのロケーションを指定してください。
 - a. sqllib/cfg ディレクトリーにある db2dj.ini ファイルを編集し、TNS_ADMIN 環境変数を次のように設定します。

```
TNS_ADMIN=x:/path/
```
 - b. DB2 インスタンスをリサイクルし、プログラムで環境変数が確実に設定されるようにするため、次のコマンドを発行します。

```
db2stop
db2start
```
3. Oracle **sqlplus** ツールを使用して接続をテストし、クライアント・ソフトウェアが Oracle サーバーに接続できることを確認してください。

この一連のタスクの次のタスクは、『Oracle ラッパーの登録』です。

関連タスク:

- 371 ページの『Oracle ラッパーの登録』
- 379 ページの『Oracle データ・ソースのための構成のチューニングとトラブルシューティング』

Oracle ラッパーの登録

Oracle ラッパーの登録は、フェデレーテッド・サーバーに Oracle データ・ソースを追加するという、より大きな作業の一部を成しています。

Oracle データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER NET8
```

推奨: NET8 というデフォルトのラッパー名を使用してください。いずれかのデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパーに関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは異なる名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターも含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で oracle_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER oracle_wrapper LIBRARY 'libdb2net8.a'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『Oracle ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連のタスクの次のタスクは、『Oracle ラッパーのサーバー定義の登録』です。

関連タスク:

- 373 ページの『Oracle データ・ソースのサーバー定義の登録』

関連資料:

- 372 ページの『Oracle ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Oracle ラッパー・ライブラリー・ファイル

次の表は、Oracle ラッパーのためのディレクトリー・パスとライブラリー・ファイル名をリストします。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは、libdb2net8.a、libdb2net8F.a、および libdb2net8U.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 82. Oracle ラッパー・ライブラリーの場所とファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2net8F.a (NET8)
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2net8F.sl (NET8)
Linux	/opt/IBM/db2/V8.1/lib	libdb2net8F.so (NET8)
Solaris	/opt/IBM/db2/V8.1/lib	libdb2net8F.so (NET8)
Windows	%DB2PATH%\%bin	db2net8.dll (NET8)

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 373 ページの『Oracle データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Oracle データ・ソースのサーバー定義の登録

Oracle データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに Oracle データ・ソースを追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスしたいそれぞれの Oracle サーバーを定義する必要があります。最初に Oracle データ・ソースのノード名を見つけてから、サーバーを登録する際にそのノード名を使用しなければなりません。

手順:

Oracle データ・ソースのサーバー定義を登録するには、次のようにします。

1. Oracle `tnsnames.ora` ファイルでノード名を見つけます。

`tnsnames.ora` ファイルの例:

```
paris_node =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = somehost)(PORT = 1521)))
    (CONNECT_DATA = (SERVICE_NAME = ora9i.seel)))
```

この例の場合、`CREATE SERVER` ステートメントで使用するノード値は `paris_node` です。

`node_name` は `CREATE SERVER SQL` ステートメントでオプションとなっていますが、Oracle データ・ソースの場合は必要です。

2. `CREATE SERVER` ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE oracle VERSION 8.1.7 WRAPPER net8
  OPTIONS (NODE 'node_name')
```

サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、`ALTER SERVER` ステートメントを使用します。

この一連のタスクの次のタスクは、『Oracle データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 375 ページの『Oracle データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「*SQL* リファレンス 第 2 巻」の『`ALTER SERVER` ステートメント』
- 「*SQL* リファレンス 第 2 巻」の『`CREATE SERVER` ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 374 ページの『`CREATE SERVER` ステートメント - Oracle ラッパーの例』

CREATE SERVER ステートメント - Oracle ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して Oracle ラッパー用サーバーを登録する方法を示す例が提供されます。必要パラメーターを使用してサーバーを登録する方法の完全指定の例、およびその他のサーバー・オプションの例がこのトピックには含まれています。

完全指定の例:

以下に示すのは、CREATE SERVER ステートメントを使用して、Oracle ラッパーのサーバー定義を登録する方法の例です。

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER net8
  OPTIONS (NODE 'paris_node')
```

oraserver

Oracle データベース・サーバーに割り当てた名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE *oracle*

アクセスを構成するデータ・ソース・サーバーのタイプを指定します。SQLNET と NET8 ラッパーのタイプ・パラメーターは *oracle* でなければなりません。

VERSION *8.1.7*

アクセスする Oracle データベース・サーバーのバージョン。サポートされる Oracle バージョンは 7.3.4、8.x、および 9.x です。

WRAPPER *net8*

CREATE WRAPPER ステートメントで指定した名前。

NODE *'paris_node'*

Oracle データベース・サーバーが存在するノードの名前。ノード名は、tnsnames.ora ファイルから入手します。

ノード名は CREATE SERVER ステートメントでオプションとなっていますが、Oracle データ・ソースの場合は必要です。

サーバー・オプションの例:

サーバー定義の作成時に、CREATE SERVER ステートメントに追加のサーバー・オプションを指定することができます。サーバー・オプションには、汎用のサーバー・オプションと Oracle に特有のサーバー・オプションがあります。

DB2 は、Oracle VARCHAR 列にはすべて、末尾ブランクが含まれていると想定しています。Oracle データベース内のすべての VARCHAR 列が末尾ブランクを含まないことが確実である場合は、データ・ソースがブランク埋め込みなしの VARCHAR 比較セマンティックを使用するということを示す、サーバー・オプションを設定できます。

以下に、サーバー・オプションを追加した Oracle サーバー定義の例を示します。

```
CREATE SERVER oraserver TYPE oracle VERSION 8.1.7 WRAPPER net8
  OPTIONS (NODE 'paris_node', VARCHAR_NO_TRAILING_BLANKS 'Y')
```

末尾ブランクを含む列が 1 つもない場合は、VARCHAR_NO_TRAILING_BLANKS サーバー・オプションを使用します。末尾ブランクを含む VARCHAR 列がいくつかわからない場合は、ALTER NICKNAME ステートメントを使用して、これらの特定の列にオプションを設定できます。

関連タスク:

- 373 ページの『Oracle データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

Oracle データ・ソースのユーザー・マッピングの作成

Oracle データ・ソースのユーザー・マッピングの作成は、Oracle データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Oracle サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスして Oracle データ・ソースへの分散要求を送信するユーザー ID ごとに、ユーザー・マッピングを作成します。

手順:

ローカル・ユーザー ID を Oracle サーバーのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR userid SERVER oraserver
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

この一連のタスクの次のタスクは、『Oracle サーバーへの接続のテスト』です。

関連タスク:

- 377 ページの『Oracle サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 375 ページの『CREATE USER MAPPING ステートメント - Oracle ラッパーの例』

CREATE USER MAPPING ステートメント - Oracle ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Oracle サーバーのユーザー ID およびパス

ワードにマップする方法を示す例が提供されます。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を Oracle サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR robert SERVER oraserver
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now')
```

robert Oracle サーバーで定義されたユーザー ID にマップする、ローカルのユーザー ID を指定します。

SERVER *oraserver*

CREATE SERVER ステートメントで定義した Oracle サーバーの名前を指定します。

REMOTE_AUTHID '*rob*'

robert をマップする Oracle データベース・サーバー側のユーザー ID を指定します。CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

REMOTE_PASSWORD '*then4now*'

'*rob*' に関連するパスワードを指定します。CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER oraserver
  OPTIONS (REMOTE_AUTHID 'rob', REMOTE_PASSWORD 'then4now')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

制約事項: Oracle データ・ソース側のユーザー ID は、Oracle **create user** コマンドで 'identified externally' 文節ではなく、'identified by' 文節を使用して作成されている必要があります。

関連タスク:

- 375 ページの『Oracle データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

Oracle サーバーへの接続のテスト

Oracle サーバーへの接続のテストは、Oracle データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Oracle サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、Oracle システム表に対して SQL SELECT ステートメントを発行します。

例:

```
SET PASSTHRU remote_server_name
SELECT count(*) FROM sys.all_tables
SET PASSTHRU RESET
```

SQL SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - Oracle サーバーを調べ、入ってくる接続に備えて構成されていることを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD のオプションが Oracle サーバーとの接続に有効な設定になっているかを確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の Oracle クライアント・ソフトウェアを調べ、Oracle サーバーに接続するように正しくインストールされて構成されていることを確認します。
 - DB2 フェデレーテッド変数の設定を調べ、Oracle サーバーに合わせた設定になっているかどうかを確認します。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
 - サーバー定義を調べます。必要であれば、それをドロップして再作成します。

この一連のタスクの次のタスクは、『Oracle 表およびビューのニックネームの登録』です。

関連タスク:

- 365 ページの『フェデレーテッド・サーバーへの Oracle データ・ソースの追加』
- 370 ページの『Oracle クライアント構成ファイルのセットアップおよびテスト』
- 378 ページの『Oracle 表およびビューのニックネームの登録』
- 366 ページの『Oracle 環境変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

Oracle 表およびビューのニックネームの登録

Oracle 表およびビューのニックネームの登録は、フェデレーテッド・サーバーに Oracle データ・ソースを追加するという、より大きな作業の一部を成しています。

定義するそれぞれの Oracle サーバーで、アクセスしたい各表またはビューごとにニックネームを登録します。Oracle サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、**CREATE NICKNAME** ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録した時に収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オブティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れられます。オブティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用するため、ニックネームを登録する前に、データ・ソース側で (DB2 の **RUNSTATS** と同等のデータ・ソース・コマンドを使用して) 統計を更新してください。

手順:

ニックネームを登録するには、**CREATE NICKNAME** ステートメントを発行します。

例:

```
CREATE NICKNAME oracle_name FOR oraserver."remote_schema"."remote.table"
```

ニックネームの長さは 128 文字までです。

ニックネームを作成したい Oracle の表またはビューごとに、このステップを繰り返します。

ニックネームを作成すると、DB2 は接続を使用してデータ・ソース・カタログを照会します。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連資料:

- 「コマンド・リファレンス」の『**RUNSTATS** コマンド』
- 「SQL リファレンス 第 2 巻」の『**CREATE NICKNAME** ステートメント』
- 378 ページの『**CREATE NICKNAME** ステートメント - Oracle ラッパーの例』

CREATE NICKNAME ステートメント - Oracle ラッパーの例

このトピックでは、**CREATE NICKNAME** ステートメントを使用して、アクセスしたい Oracle の表またはビューのニックネームを登録する方法の例が提供されます。

この例では、ニックネームが割り当てられる Oracle サーバーのリモート・オブジェクトを指定する方法が示されます。

```
CREATE NICKNAME PARISINV FOR oraserver."france"."inventory"
```

PARISINV

Oracle の表またはビューを識別するために使用する固有なニックネーム。

注: ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの登録時にスキーマを省略すると、そのニックネームのスキーマはニックネームを登録するユーザーの許可 ID になります。

```
oraserver."france"."inventory"
```

リモート・オブジェクトを表す、3 つの部分からなる ID。

- *oraserver* は、CREATE SERVER ステートメントで Oracle データベース・サーバーに割り当てた名前です。
- *france* は、表またはビューが属するリモート・スキーマの名前です。
- *inventory* は、アクセスしたいリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、名前を引用符で囲んでいない限り、Oracle スキーマおよび表の名前を大文字に変換します。

関連タスク:

- 378 ページの『Oracle 表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

Oracle データ・ソースのための構成のチューニングとトラブルシューティング

Oracle データ・ソースに対する構成をセットアップした後でも、パフォーマンスを改善するために構成を変更することができます。

接続の問題

tnsnames.ora ファイルの DESCRIPTION セクションにある HOST ごとに、TCP/IP hosts ファイルを更新する必要がある場合があります。このファイルを更新するかどうかは、ネットワークで TCP/IP がどのように構成されているかによります。ネットワークの一部で、tnsnames.ora ファイルの DESCRIPTION セクションに指定されたリモート・ホスト名をアドレスに変換する必要があります。

ホスト名を認識するネーム・サーバーがネットワークにある場合は、TCP/IP hosts ファイルを更新する必要はありません。そうでない場合は、リモート・ホストのための項目が必要です。ネットワークがどのように構成されているかについては、ネットワーク管理者に問い合わせてください。hosts ファイルを更新する必要がある場合、ファイルの場所は、フェデレーテッド・サーバーのオペレーティング・システムにより異なります。

UNIX が稼働するフェデレーテッド・サーバーの場合

/etc/hosts ファイルを更新します。

Windows が稼働するフェデレーテッド・サーバーの場合

x:%winnt%system32\drivers\etc\hosts ファイルを更新します。

関連タスク:

- 365 ページの『フェデレーテッド・サーバーへの Oracle データ・ソースの追加』

関連資料:

- 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』

第 20 章 Sybase データ・ソースへのアクセスの構成

この章では、Sybase データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Sybase データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

Sybase のフェデレーテッド・システムへの追加

フェデレーテッド・サーバーへの Sybase データ・ソースの追加

フェデレーテッド・サーバーが Sybase データ・ソースにアクセスするように構成するには、アクセスしたいデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供する必要があります。

フェデレーテッド・サーバーを Sybase データ・ソースにアクセスするように構成するには、DB2 コントロール・センターまたは DB2 コマンド行を使用できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

フェデレーテッド・サーバーに Sybase データ・ソースを追加するには、次のようにします。

1. Sybase クライアント構成ファイルをセットアップしてテストします。
2. Sybase 環境変数を設定します。
3. ラッパーを登録します。
4. サーバー定義を登録します。
5. ユーザー・マッピングを作成します。
6. Sybase サーバーとの接続をテストします。
7. Sybase の表およびビューのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』

- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 384 ページの『Sybase クライアント構成ファイルのセットアップおよびテスト』
- 386 ページの『Sybase ラッパーの登録』
- 387 ページの『Sybase データ・ソースのサーバー定義の登録』
- 391 ページの『Sybase データ・ソースのユーザー・マッピングの作成』
- 392 ページの『Sybase サーバーへの接続のテスト』
- 393 ページの『Sybase 表およびビューのニックネームの登録』
- 395 ページの『Sybase データ・ソースに対する構成のチューニングおよびトラブルシューティング』
- 382 ページの『Sybase 環境変数の設定』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Sybase 環境変数の設定

Sybase 環境変数の設定は、フェデレーテッド・サーバーに Sybase を追加するという、より大きなタスクの一部を成しています。

DB2 Information Integrator をインストールする際、インストール・プロセスにおいて、db2dj.ini ファイルでの Sybase 環境変数の設定が試行されます。

以下の場合には、環境変数は db2dj.ini ファイルに設定されません。

- フェデレーテッド・サーバーのセットアップ後に Sybase クライアント・ソフトウェアをインストールする場合。
- Sybase クライアント・ソフトウェアをインストールしていない場合。

Sybase の有効な環境変数は次のとおりです。

- SYBASE
- SYBASE_OCS (Sybase バージョン 12 以降で必須)
- SYBASE_CHARSET (オプション)

オプション環境変数は手動で設定する必要があります。

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。 『db2dj.ini ファイルの制約事項』

手順:

必要な環境変数を自動的に設定するには、次のようにします。

1. まだインストールされていない場合、フェデレーテッド・サーバーにクライアント・ソフトウェアをインストールし、構成します
2. 必要な環境変数を設定します。DB2 Information Integrator のインストールを再実行することによって、環境変数を自動的に設定できます。ランチパッドから「製品のインストール」をクリックし、ウィザードの指示に従います。

環境変数を手動で設定するには、次のようにします。

1. db2dj.ini ファイルを編集します。
 - Windows が稼働するフェデレーテッド・サーバーの場合、このファイルは `sql1lib¥cfg` ディレクトリーにあります。
 - UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは `sql1lib/cfg` ディレクトリーにあります。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Sybase クライアント・ソフトウェアに関する構成情報が入っています。このファイルが存在しない場合、任意のテキスト・エディターを使用して、db2dj.ini という名前の新規ファイルを作成できます。db2dj.ini ファイルには、変数に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。

2. 必要に応じて以下の環境変数を設定します。

SYBASE

SYBASE 環境変数を、Sybase Open Client ソフトウェアがインストールされているディレクトリー・パスに設定します。この変数の完全修飾パスを指定します。たとえば、次のようにします。

```
SYBASE=sybase_home_directory
```

たとえば、ディレクトリー・パスが `D:/djxclient/sybase/V12/OCS-12_5` の場合、次の SYBASE 変数を指定します。

```
SYBASE=D:/djxclient/sybase/V12
```

SYBASE_OCS

Sybase Open Client バージョン 12 以降の場合、SYBASE_OCS 環境変数を OCS ディレクトリーの名前に設定します。完全修飾パスを指定してはなりません。SYBASE_OCS 環境変数は、インストールされている Sybase Open Client のバージョンとリリースを指定します。

```
SYBASE_OCS=OCS-version_release
```

たとえば、ディレクトリー・パスが `D:/djxclient/sybase/V12/OCS-12_5` の場合、SYBASE_OCS 変数に次のディレクトリーを指定します。

```
SYBASE_OCS=OCS-12_5
```

SYBASE_CHARSET

SYBASE_CHARSET 変数に、使用する文字セットの名前を設定します。Sybase ラッパーは、使用すべき文字セットの判別に SYBASE_CHARSET を使用します。SYBASE_CHARSET 変数が設定されない場合、ラッパーはフェデレーテッド・サーバーの文字セットで指定したのと同じ Sybase 文字セットを使用します。一致する Sybase 文字セットがない場合、ラッパーは iso_1 文字セットを使用します。\$SYBASE¥charsets ディレクトリーには、有効な文字セット名のリストがあります。たとえば、次のような項目があります。

```
SYBASE_CHARSET=iso_1
```

iso_1 は、使用する文字セットの名前です。

3. フェデレーテッド・データベース・インスタンスの .profile ファイルを SYBASE 環境変数で更新します。次のコマンドを発行すると、これを行えます。

```
export SYBASE=<sybase_home_directory>
export SYBASE_OCS=OCS-version_release
export PATH=$SYBASE/bin:$PATH
```

4. ホーム・ディレクトリーから次の入力して、フェデレーテッド・データベース・インスタンス .profile を実行します。

```
..profile
```

5. フェデレーテッド・サーバーに環境変数を設定するには、フェデレーテッド・データベース・インスタンスをリサイクルします。フェデレーテッド・データベース・インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop
db2start
```

この一連の作業における次のタスクは『Sybase ラッパーの登録』です。

関連タスク:

- 393 ページの『Sybase 表およびビューのニックネームの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』
- 66 ページの『db2dj.ini ファイルの制約事項』

Sybase クライアント構成ファイルのセットアップおよびテスト

Sybase クライアント構成ファイルのセットアップとテストは、Sybase データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

クライアント構成ファイルは、フェデレーテッド・サーバーにインストールされている Sybase Open Client ライブラリーを使用して、Sybase に接続するために使用されます。このファイルは、各 Sybase SQL Server および Adaptive Server Enterprise インスタンスのロケーションと、データベース・サーバーへの接続のタイプ (プロトコル) を指定します。

Sybase に接続するために使用されるフェデレーテッド・サーバー内のそれぞれのインスタンスごとに、クライアント構成ファイルをセットアップする必要があります。

す。このファイルのセットアップおよびテストに必要なステップは、フェデレーテッド・サーバーで稼働しているオペレーティング・システムによって異なります。

手順:

クライアント構成ファイルをセットアップし、テストするには、次のようにします。

UNIX オペレーティング・システムの場合:

1. Sybase Open Client ソフトウェアに付属のユーティリティーを使用して、クライアント構成ファイルをセットアップします。このファイルは \$SYBASE ディレクトリに作成されます。ファイルの名前は interfaces です。このユーティリティーの使用法の詳細は、Sybase の資料を参照してください。
2. 次のいずれかの方法で、interfaces ファイルをフェデレーテッド・データベースのインスタンスに対してアクセス可能にしてください。
 - このファイルをフェデレーテッド・データベース・インスタンスの \$HOME/sql1lib ディレクトリにコピーします。
 - **ln** コマンドを使用して、\$sybase ディレクトリからインスタンスの \$HOME/sql1lib ディレクトリ内の interfaces ファイルへのリンクを作成します。たとえば、次のようにします。

```
ln -s -f /home/sybase/interfaces /home/db2dinst1/sql1lib
```
 - **IFILE** サーバー・オプションを使用して、Sybase の interfaces ファイルへの絶対パスを指定します。
3. Sybase Open クライアント・ソフトウェアが Sybase サーバーに接続できるか確認するために、接続をテストしてください。テストには、該当の Sybase 照会ユーティリティー (**isql** など) を使用してください。たとえば、次のようにします。

```
isql -Ssybnode -Umary -I/home/db2dinst1/sql1lib/interfaces
```

Windows オペレーティング・システムの場合:

1. Sybase Open Client ソフトウェアに付属のユーティリティーを使用して、クライアント構成ファイルをセットアップします。このファイルは、%SYBASE%\ini ディレクトリに作成されます。ファイルの名前は sql.ini です。このユーティリティーの使用法の詳細は、Sybase の資料を参照してください。
2. この sql.ini ファイルをフェデレーテッド・データベース・インスタンスの %DB2PATH% ディレクトリにコピーし、このファイルをフェデレーテッド・データベース・インスタンスに対してアクセス可能にします。デフォルト・パスは、c:%Program Files%\IBM\SQLLIB です。

DB2 Information Integrator は、Sybase クライアント構成ファイルのデフォルト名として interfaces を使用するため、Windows では、c:%Program Files%\IBM\SQLLIB ディレクトリにある sql.ini ファイルを interfaces に名前変更します。

必須: sql.ini ファイルを interfaces という名前に変更しない場合は、サーバー定義の作成時に **IFILE** サーバー・オプションを使用する必要があります。

3. Sybase Open クライアント・ソフトウェアが Sybase サーバーに接続できるか確認するために、接続をテストしてください。テストには、該当の Sybase 照会ユーティリティ (isql など) を使用してください。たとえば、次のようにします。

```
isql -Ssybnode -Umary -I"c:\Program Files\IBM\SQLLIB\interfaces"
```

この一連の作業における次のタスクは『Sybase ラッパーの登録』です。

関連タスク:

- 386 ページの『Sybase ラッパーの登録』

Sybase ラッパーの登録

Sybase ラッパーの登録は、フェデレーテッド・サーバーに Sybase データ・ソースを追加するという、より大きな作業の一部を成しています。

Sybase データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

たとえば、次のようにします。

```
CREATE WRAPPER CTLIB
```

推奨事項: CTLIB というデフォルトのラッパー名を使用してください。このデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパー名に関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上で sybase_wrapper という名前を使用してラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER sybase_wrapper LIBRARY 'libdb2ctl.a'
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『Sybase ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは『Sybase ラッパーのサーバー定義の登録』です。

関連タスク:

- 387 ページの『Sybase データ・ソースのサーバー定義の登録』

関連資料:

- 387 ページの『Sybase ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Sybase ラッパー・ライブラリー・ファイル

以下の表は、Sybase ラッパーのディレクトリー・パスとライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb2ctlib.a、libdb2ctlib.f.a、およびlibdb2ctlibU.a のラッパー・ライブラリー・ファイルが追加されます。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 83. Sybase ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2ctlib.a (CTLIB)
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2ctlib.sl (CTLIB)
Linux	/opt/IBM/db2/V8.1/lib	libdb2ctlib.so (CTLIB)
Solaris	/opt/IBM/db2/V8.1/lib	libdb2ctlib.so (CTLIB)
Windows	%DB2PATH%\%bin	db2ctlib.dll (CTLIB)

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 386 ページの『Sybase ラッパーの登録』

Sybase データ・ソースのサーバー定義の登録

Sybase データ・ソースのサーバー定義の登録は、フェデレーテッド・サーバーに Sybase データ・ソースを追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスしたいそれぞれの Sybase サーバーを定義する必要があります。

制約事項:

DB2 コントロール・センターを使用して Sybase データ・ソースのサーバー定義を登録するには、Sybase クライアント構成ファイル `interfaces` をデフォルト・ディレクトリーに置く必要があります。

- UNIX を実行するフェデレーテッド・サーバーの場合、デフォルト・ディレクトリーは `$HOME/sql1lib/` です。
- Windows を実行するフェデレーテッド・サーバーの場合、デフォルト・ディレクトリーは `c:\Program Files\IBM\SQLLIB` です。

手順:

サーバー定義は、DB2 コントロール・センターまたは DB2 コマンド行から登録できます。

- このタスクを DB2 コントロール・センターから実行するには、「フェデレーテッド・オブジェクト (Federated Objects)」ウィザードを使用するか、あるいは「サーバー定義 (Server Definitions)」フォルダーを右マウス・ボタンでクリックして、「作成」をクリックします。発見ツールを使用すると、Sybase サーバーのノード名を検索できます。サーバー定義を登録するには、DBNAME サーバー・オプションに情報を指定する必要があります。
- このタスクを DB2 コマンド行から行うには、CREATE SERVER ステートメントを使用します。

1. Sybase `interfaces` ファイルでノード名を見つけます。

UNIX オペレーティング・システムの場合の `interfaces` ファイルの例:

```
sybase119
query tcp ether anaconda 4100
```

Windows NT または Windows 2000 オペレーティング・システムの場合の `interfaces` ファイルの例

```
[sybase119]
query=TCP,anaconda,4100
```

これらの例では、ノード名は `sybase119` です。このノード名の後に、接続のタイプ (TCP/IP) とホスト名 (`anaconda`) が続きます。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

2. CREATE SERVER ステートメントを発行します。

たとえば、次のようにします。

```
CREATE SERVER server_name TYPE SYBASE
VERSION 12.0 WRAPPER CTLIB
OPTIONS (NODE 'sybnode', DBNAME 'sybdb')
```

サーバー定義を作成した後、サーバー・オプションを追加またはドロップするには、ALTER SERVER ステートメントを使用します。

重要: Sybase クライアント構成ファイルをセットアップする際に `sql.ini` ファイルを `interfaces` という名前に変更しない場合には、サーバー定義の登録時に IFILE サーバー・オプションを使用する必要があります。

この一連の作業における次のタスクは『Sybase データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 391 ページの『Sybase データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』
- 389 ページの『CREATE SERVER ステートメント - Sybase ラッパーの例』

CREATE SERVER ステートメント - Sybase ラッパーの例

このトピックでは、CREATE SERVER ステートメントを使用して Sybase ラッパー用のサーバー定義を登録する方法の例が示されます。必要パラメーターを使用してサーバー定義を登録する方法の完全指定の例、およびその他のサーバー・オプションの例がこのトピックには含まれています。

完全指定の例:

次に、CREATE SERVER ステートメントを実行して Sybase ラッパーのサーバー定義を登録する方法の例を示します。

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB  
OPTIONS (NODE 'sybnode', DBNAME 'sybdb')
```

SYBSERVER

Sybase サーバーに割り当てる名前。この名前は固有でなければなりません。重複するサーバー名は使用できません。

TYPE SYBASE

アクセスを構成しているデータ・ソースのタイプとして Sybase を指定します。CTLIB ラッパーの TYPE パラメーターは、SYBASE でなければなりません。

VERSION 12.0

アクセスしたい Sybase データベース・サーバー・ソフトウェアのバージョン。サポートされるバージョンは 11、11.5、11.9、12、および 12.5 です。

WRAPPER CTLIB

CREATE WRAPPER ステートメントで指定したラッパー名。

NODE 'sybnode'

SYBSERVER があるノードの名前。ノード名は、interfaces ファイルにあります。この値は大文字小文字の区別があります。

ノード名は CREATE SERVER ステートメントではオプションとなっていますが、Sybase データ・ソースの場合は必要です。

DBNAME 'sybdb'

アクセスしたい Sybase データベースの名前。この名前は Sybase サーバーにあります。この値は大文字小文字の区別があります。

データベース名は CREATE SERVER ステートメントではオプションとな
っていますが、 Sybase データ・ソースの場合は必要です。

重要: Sybase クライアント構成ファイルをセットアップする際に sql.ini ファイル
を interfaces という名前に変更しない場合には、サーバー定義の登録時に IFILE
サーバー・オプションを使用する必要があります。

サーバー・オプションの例:

サーバーを登録する際に、CREATE SERVER ステートメントに追加のサーバー・オ
プションを指定することができます。こうしたサーバー・オプションには、汎用の
サーバー・オプションと Sybase 特有のサーバー・オプションがあります。

次に、UNIX を実行するフェデレーテッド・サーバー上でサーバー定義を登録する
場合に、TIMEOUT サーバー・オプションを使用する方法の例を示します。

```
|  
| CREATE SERVER SYBSERVER TYPE SYBASE  
|         VERSION 12.0 WRAPPER CTLIB  
|         OPTIONS (NODE 'sybnode', DBNAME 'sybdb',  
|                 TIMEOUT '60')
```

タイムアウト値は、ラッパーが Sybase サーバーからの応答を待つ秒数です。
TIMEOUT オプションを使用して、トランザクションのデッドロックを避けてくだ
さい。

次に、Windows を実行するフェデレーテッド・サーバー上でサーバー定義を登録す
る場合に、IFILE サーバー・オプションを使用する方法の例を示します。

```
|  
| CREATE SERVER SYBSERVER TYPE SYBASE  
|         VERSION 12.0 WRAPPER CTLIB  
|         OPTIONS (NODE 'sybnode', DBNAME 'sybdb',  
|                 IFILE 'C:¥Sybase¥ini¥sql.ini')
```

IFILE 値は、Sybase Open Client interfaces ファイルの絶対パスと名前です。
sql.ini ファイルを、\$SQLLIB¥interfaces (UNIX システムの場合) または
%SQLLIB%/interfaces (Windows オペレーティング・システムの場合) という名前
でコピーあるいはリンクしていない場合には、このサーバー・オプションを使用し
てください。

その他の Sybase 特有のサーバー・オプションは、以下のとおりです。

- LOGIN_TIMEOUT
- PACKET_SIZE

関連タスク:

- 387 ページの『Sybase データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』

Sybase データ・ソースのユーザー・マッピングの作成

Sybase データ・ソースのユーザー・マッピングの作成は、Sybase データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Sybase サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Sybase データ・ソースへの分散要求を送信します。

手順:

ローカル・ユーザー ID を Sybase サーバーのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

たとえば、次のようにします。

```
CREATE USER MAPPING FOR userid SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

この一連の作業における次のタスクは『Sybase サーバーへの接続のテスト』です。

関連タスク:

- 392 ページの『Sybase サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 391 ページの『CREATE USER MAPPING ステートメント - Sybase ラッパーの例』

CREATE USER MAPPING ステートメント - Sybase ラッパーの例

このトピックでは、CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Sybase サーバーのユーザー ID およびパスワードにマップする方法を示す例が提供されます。このトピックには、必要パラメーターを使用する完全指定の例と、CREATE USER MAPPING ステートメントで DB2 特殊レジスター USER を使用する方法を示す例が含まれます。

完全指定の例:

以下に示すのは、フェデレーテッド・サーバーのユーザー ID を Sybase サーバーのユーザー ID およびパスワードにマップする方法を示す例です。

```
CREATE USER MAPPING FOR maria SERVER SYBSERVER
    OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night')
```

maria Sybase サーバーで定義されたユーザー ID にマップするローカルのユーザー ID を指定します。

SERVER SYBSERVER

CREATE SERVER ステートメントで定義した Sybase サーバーの名前を指定します。

REMOTE_AUTHID 'mary'

maria をマップする Sybase サーバー側のユーザー ID を指定します。

CREATE SERVER ステートメントで FOLD_ID サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

REMOTE_PASSWORD 'day2night'

'mary' に関連するパスワードを指定します。 CREATE SERVER ステートメントで FOLD_PW サーバー・オプションを 'U' または 'L' にしている場合を除き、この値の大文字小文字の区別を保持するように単一引用符を使用してください。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER SYBSERVER
  OPTIONS (REMOTE_AUTHID 'mary', REMOTE_PASSWORD 'day2night')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 391 ページの『Sybase データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』

Sybase サーバーへの接続のテスト

Sybase サーバーへの接続のテストは、Sybase データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Sybase サーバーへの接続のテストは、定義したサーバー定義とユーザー・マッピングを使用して行うことができます。

手順:

接続をテストするには、次のようにします。

1. パススルー・セッションを開き、Sybase システム表に対して SQL SELECT ステートメントを発行します。

たとえば、次のようにします。

```
| SET PASSTHRU remote_server_name  
| SELECT count(*) FROM dbo.sysobjects  
| SET PASSTHRU RESET
```

| *remote_server_name* は、CREATE SERVER ステートメントにサーバー定義を登録した際にリモート・サーバーに指定した名前です。SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - Sybase サーバーを調べ、入ってくる接続に備えて構成されていることを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID と REMOTE_PASSWORD のオプションが Sybase サーバーとの接続に有効な設定になっているかを確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の Sybase クライアント・ソフトウェアをチェックし、これが正しくインストールされ、Sybase サーバーに接続するように正しく構成されていることを確認します。
 - DB2 フェデレーテッド変数の設定を調べ、Sybase サーバーに合わせた設定になっているかどうかを確認します。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
 - サーバー定義を調べます。必要であれば、それをドロップして再作成します。

この一連の作業における次のタスクは『Sybase 表およびビューのニックネームの登録』です。

関連タスク:

- 381 ページの『フェデレーテッド・サーバーへの Sybase データ・ソースの追加』
- 384 ページの『Sybase クライアント構成ファイルのセットアップおよびテスト』
- 387 ページの『Sybase データ・ソースのサーバー定義の登録』
- 393 ページの『Sybase 表およびビューのニックネームの登録』
- 382 ページの『Sybase 環境変数の設定』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER USER MAPPING ステートメント』

Sybase 表およびビューのニックネームの登録

Sybase 表およびビューのニックネームの登録は、フェデレーテッド・サーバーに Sybase データ・ソースを追加するという、より大きな作業の一部を成しています。

定義するそれぞれの Sybase サーバーで、アクセスしたい各表またはビューごとにニックネームを登録します。Sybase サーバーを照会するときには、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、**CREATE NICKNAME** ステートメントを使用してデータ・ソース・オブジェクトにニックネームを登録した時に収集されます。フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するかチェックし、次に既存のデータ・ソース統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れます。オプティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用するため、ニックネームを登録する前に、データ・ソース側で (**DB2** の **RUNSTATS** と同等のデータ・ソース・コマンドを使用して) 統計を更新してください。

手順:

ニックネームを登録するには、**CREATE NICKNAME** ステートメントを発行します。

たとえば、次のようにします。

```
CREATE NICKNAME sybase_name FOR SYBSERVER."remote_schema"."remote.table"
```

ニックネームの長さは 128 文字までです。

ニックネームを作成したい Sybase の表またはビューごとに、このステップを繰り返します。

ニックネームを作成すると、フェデレーテッド・サーバーは指定の情報を使用してデータ・ソース・カタログを照会します。この照会は、ニックネームを使用してデータ・ソースへの接続をテストします。接続が働かない場合、エラー・メッセージが出ます。

関連資料:

- 「コマンド・リファレンス」の『**RUNSTATS** コマンド』
- 「**SQL** リファレンス 第 2 巻」の『**CREATE NICKNAME** ステートメント』
- 394 ページの『**CREATE NICKNAME** ステートメント - Sybase ラッパーの例』

CREATE NICKNAME ステートメント - Sybase ラッパーの例

このトピックでは、**CREATE NICKNAME** ステートメントを使用して、アクセスしたい Sybase の表およびビューのニックネームを登録する方法の例が示されます。

この例は、ニックネームを割り当てた、Sybase サーバーのリモート・オブジェクトを指定する方法を示しています。

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."salesdata"."europe"
```

SYBSALES

これは、Sybase の表またはビューの固有なニックネームです。

ニックネームは、2 つの部分 (スキーマとニックネーム) からなる名前です。ニックネームの作成時にスキーマを省略すると、そのニックネームのスキーマはニックネームを作成したユーザーの認証 ID になります。

SYBSERVER."salesdata"."europe"

リモート・オブジェクトを表す、3つの部分からなる ID です。

- *SYBSERVER* は、CREATE SERVER ステートメントで Sybase データベース・サーバーに割り当てた名前です。
- *salesdata* は、表またはビューが属するリモート・スキーマの名前です。
- *europe* は、アクセスしたいリモートの表またはビューの名前です。

フェデレーテッド・サーバーは、引用符で名前を囲まない限り、Sybase スキーマおよび表の名前を大文字に変換します。

関連タスク:

- 393 ページの『Sybase 表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE NICKNAME ステートメント』

Sybase データ・ソースに対する構成のチューニングおよびトラブルシューティング

Sybase データ・ソースに対する構成をセットアップした後、パフォーマンスを改善するために構成を変更したい場合があります。

sp_helpindex エラーを解決する

フェデレーテッド・システムは Sybase カタログ・ストアード・プロシージャの 1 つである、sp_helpindex を必要とします。次の SQL エラーが出た場合、Sybase カタログ・ストアード・プロシージャが Sybase サーバーにインストールされていない可能性があります。

```
SQL0204N "sp_helpindex" is an undefined name.
```

カタログ・ストアード・プロシージャを Sybase サーバーにインストールするように、Sybase 管理者に依頼してください。

関連タスク:

- 381 ページの『フェデレーテッド・サーバーへの Sybase データ・ソースの追加』

関連資料:

- 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』

第 21 章 表構造ファイル・データ・ソースへのアクセスの構成

この章では、表構造ファイル・データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。表構造ファイル・データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- 表構造ファイルの概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- 表構造ファイル・ラッパーに関連するエラー・メッセージをリストします。

表構造ファイルとは ?

表構造のファイルは通常、一連のレコードによって構成されます。各レコードには同じ数のフィールドがあり、任意の区切り文字によって区切られています。 NULL 値は 2 つの区切り文字が続くことによって表されます。

以下の例は、DRUGDATA1.TXT というファイルの内容を示しています。このファイルは、3 つのレコードから構成されており、それぞれのレコードにコンマによって区切られている 3 つのフィールドがあります。

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

最初のフィールドは薬品 (drug) の固有な ID 番号です。2 番目のフィールドは薬品 (drug) の名前です。3 番目のフィールドは薬品の製造会社 (manufacturer) の名前です。

関連概念:

- 397 ページの『表構造ファイルの属性』
- 398 ページの『DB2 Information Integrator が表構造ファイルを処理する方法』

関連タスク:

- 400 ページの『フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加』

表構造ファイルの属性

表構造のファイルは、ソートされたものでもソートされていないものでもかまいません。表構造ファイル・ラッパーは、ソートされていないデータ・ファイルよりもソートされたデータ・ファイルの方が、効率よく検索できます。

ソート・ファイル

DRUGDATA1.TXT にはソートされたレコードが入っています。このファイルは、最初のフィールドである薬品の固有 ID 番号によってソートされています。このフィールドは、ドラッグごとに固有であるため主キーです。ソートされたファイルは必ず昇順でソートされていなければなりません。

```
234,DrugnameA,Manufacturer1
332,DrugnameB,Manufacturer2
333,DrugnameC,Manufacturer2
```

ソートされていないファイル

DRUGDATA2.TXT にはソートされていないレコードが入っています。ファイルにリストされているレコードの順番に規則性はありません。

```
332,DrugnameB,Manufacturer2
234,DrugnameA,Manufacturer1
333,DrugnameC,Manufacturer2
```

関連概念:

- 397 ページの『表構造ファイルとは ?』
- 398 ページの『DB2 Information Integrator が表構造ファイルを処理する方法』

関連タスク:

- 400 ページの『フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加』

DB2 Information Integrator が表構造ファイルを処理する方法

DB2[®] Information Integrator は、ラッパーと呼ばれるモジュールを使用することによって、表構造のファイルのデータを照会する SQL ステートメントを処理して、通常のリレーショナル表またはビューにあるデータのように扱うことができます。これにより、表構造のファイルにあるデータを、リレーショナル・データまたは他の表構造のファイルにあるデータと結合することができます。このプロセスは 399 ページの図 27 に図示されています。

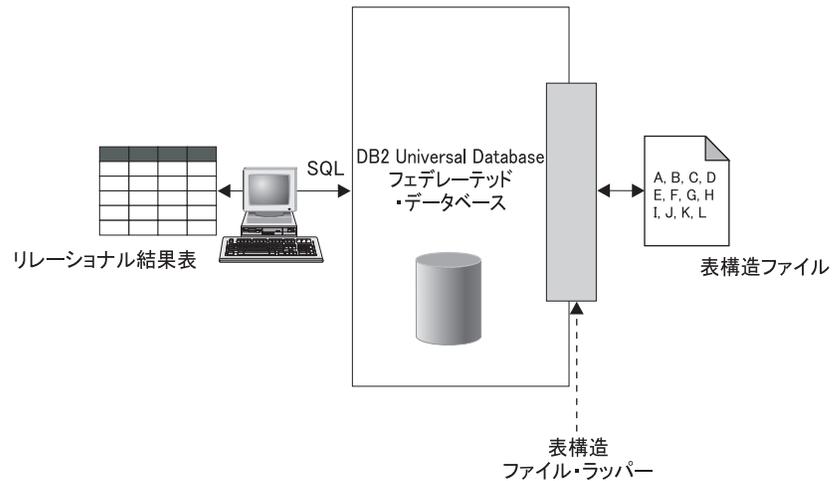


図 27. 表構造ファイル・ラッパーがどのように機能するか

たとえば、表構造のファイル、DRUGDATA2.TXT が研究所のコンピューターにあるとします。このデータを照会して、使用する他のデータ・ソースの表との突き合わせを行うことは、単調で面倒な作業となります。

DB2 Information Integrator に DRUGDATA2.TXT を登録すると、ファイルはリレーショナル・データ・ソースのようになります。その結果、ファイルを他のリレーショナル・データ・ソースおよび非リレーショナル・データ・ソースとともに照会することができ、データを同時に分析することができるようになります。

たとえば、以下に示す照会を実行するとします。

```
SELECT * FROM DRUGDATA2 ORDER BY DCODE
```

この照会は以下のような結果を生成します。

Dcode	Drug	製造メーカー
234	DrugnameA	Manufacturer1
332	DrugnameB	Manufacturer2
333	DrugnameC	Manufacturer2

関連概念:

- 397 ページの『表構造ファイルとは ?』
- 397 ページの『表構造ファイルの属性』

関連タスク:

- 400 ページの『フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加』

フェデレーテッド・システムへの表構造ファイルの追加

フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加

フェデレーテッド・サーバーが表構造ファイルのデータ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、フェデレーテッド・サーバーが表構造ファイルのデータ・ソースにアクセスするように構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

表構造のファイルのデータ・ソースをフェデレーテッド・サーバーに追加するには、以下のようにします。

1. ラッパーを登録します。
2. サーバー定義を登録します。
3. 表構造のファイルのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 400 ページの『表構造ファイル・ラッパーの登録』
- 402 ページの『表構造ファイルのサーバーの登録』
- 402 ページの『表構造ファイルのニックネームの登録』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

表構造ファイル・ラッパーの登録

表構造ファイル・ラッパーの登録は、表構造ファイルのデータ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

表構造ファイルのデータ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上にラッパーを laboratory_flat_files の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER laboratory_flat_files LIBRARY 'libdb2lsfile.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『表構造ファイル・ラッパーのライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『表構造ファイルのサーバー定義の登録』です。

関連資料:

- 401 ページの『表構造ファイル・ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

表構造ファイル・ラッパー・ライブラリー・ファイル

以下の表は、表構造ラッパーのディレクトリー・パスとライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb2lsfile.a、libdb2lsfileF.a、および libdb2lsfileU.a のラッパー・ライブラリー・ファイルが追加されます。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 84. 表構造ファイル・クライアント・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lsfile.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2lsfile.sl
Linux	/opt/IBM/db2/V8.1/lib	libdb2lsfile.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lsfile.so
Windows	%DB2PATH%¥bin	db2lsfile.dll

| %DB2PATH% は、Windows にインストールされている DB2 Information Integrator
| のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows
| ディレクトリー・パスは、C:\Program Files\IBM\SQLLIB です。

| **関連タスク:**

- | • 400 ページの『表構造ファイル・ラッパーの登録』

表構造ファイルのサーバーの登録

表構造ファイルのサーバーの登録は、表構造ファイルをフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。ラッパーが登録された後、対応するサーバーを登録する必要があります。

手順:

表構造のファイル・サーバーをフェデレーテッド・システムに登録するには、CREATE SERVER ステートメントを使用します。例:

```
CREATE SERVER biochem_lab WRAPPER laboratory_flat_files
```

この例では、biochem_lab は表構造のファイル・サーバーに割り当てられる名前です。名前は、サーバーが登録されるデータベースで固有でなければなりません。

この一連の作業における次のタスクは、『表構造ファイルのニックネームの登録』です。

関連タスク:

- 402 ページの『表構造ファイルのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

表構造ファイルのニックネームの登録

表構造ファイルのニックネームの登録は、表構造ファイルをフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。サーバーを登録した後、対応するニックネームを登録する必要があります。ニックネームは、照会の中で表構造のファイル・データ・ソースを参照する場合に使用されます。

ニックネームは、次の 2 通りの方法のいずれかで、表構造ファイルに関連付けられます。

- FILE_PATH ニックネーム・オプションを使用して、固定された方法で。このオプションが使用される場合、ニックネームは特定の表構造ファイルからのデータを表します。
- DOCUMENT ニックネーム列オプションを使用して、照会時に指定されたファイル名で。このオプションが使用される場合、ニックネームは、スキーマがニックネーム定義に一致する任意の表構造ファイルからのデータを表すために使用できます。

制約事項:

非数値フィールドがその列タイプに対して長すぎる場合、超過したデータは切り捨てられます。ファイルの 10 進数フィールドで、基数 char の後ろに列タイプの scale パラメーターで許可されているよりも多くの列がある場合、超過したデータは切り捨てられます。基数文字は、LC_NUMERIC 各国語サポート区分の RADIXCHAR 項目によって区切られます。

行の最大長は 10 MB (10485760 バイト) です。

手順:

ニックネームを登録するには、アクセスしたい表構造のファイルごとに CREATE NICKNAME を使用します。

一連の作業はこれで終わりです。

関連タスク:

- 400 ページの『フェデレーテッド・サーバーへの表構造ファイル・データ・ソースの追加』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 404 ページの『表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル』
- 405 ページの『表構造ファイル・ラッパーに関する最適化のヒントと考慮事項』
- 605 ページの『CREATE NICKNAME ステートメント構文 - 表構造ファイル・ラッパー』
- 403 ページの『CREATE NICKNAME ステートメント - 表構造ファイル・ラッパーの例』

CREATE NICKNAME ステートメント - 表構造ファイル・ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して、表構造ファイル・ラッパーのニックネームを登録する完全指定の例が示されます。また、特定のオプションの例も含まれています。

完全指定の例:

次に、表構造のファイル DRUGDATA1.TXT に関する CREATE NICKNAME ステートメントの例を示します。

```
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')
```

KEY COLUMN オプションの例:

次に、NOT NULL オプションをニックネーム・ステートメントの中にその定義に追加することによって、キーとして指定された列を NULL 不可能に指定する例を示します。

```
CREATE NICKNAME tox (tox_id INTEGER NOT NULL, toxicity VARCHAR(100))
FOR SERVER tox_server1
  OPTIONS (FILE_PATH'/tox_data.txt', SORTED 'Y')
```

```
CREATE NICKNAME weights (mol_id INTEGER, wt VARCHAR(100) NOT NULL)
FOR SERVER wt_server
  OPTIONS (FILE_PATH'/wt_data.txt', SORTED 'Y', KEY_COLUMN 'WT')
```

このオプションには大文字小文字の区別があります。ただし DB2 は、列が二重引用符によって定義されていない限り、列名を大文字に変換します。以下の例は正しく処理されません。なぜなら、empno 列は DB2 によって大文字に変換されますが、empno キー列は小文字でサブミットされるからです。したがって、キーとして指定された列は検出されません。

```
CREATE NICKNAME depart (
  empno char(6) NOT NULL)
FOR SERVER DATASTORE
  OPTIONS(FILE_PATH'data.txt', SORTED 'Y', KEY_COLUMN 'empno');
```

関連タスク:

- 402 ページの『表構造ファイルのニックネームの登録』

関連資料:

- 605 ページの『CREATE NICKNAME ステートメント構文 - 表構造ファイル・ラッパー』

表構造ファイル・ラッパーのファイル・アクセス・コントロール・モデル

データベース管理システムは、DB2 インスタンス所有者の権限によって表構造ファイルにアクセスします。ラッパーは、このユーザー ID (またはグループ ID) によって読み取ることができるファイルにだけアクセスできます。アプリケーションの許可 ID (フェデレーテッド・データベースへの接続を確立する ID) は関係ありません。

DB2 Universal Database Enterprise Server Edition では、ニックネームが作成されている表構造のファイルはすべて、各ノードから同じパス名を使用してアクセス可能でなければなりません。ファイルは、すべてのノードから共通のパスを使用してアクセス可能であれば、DB2 Universal Database のノードに置く必要はありません。

ネットワーク内に Windows NT または Windows 2000 ドメインが構成されている場合に、マップ・ドライブ上の表構造ファイルにアクセスするためには、DB2 サービス・ログオン・アカウントが、データ・ファイルの存在する共有フォルダーへのアクセス権が付与されたドメインからのアカウントでなければなりません。

ネットワークに Windows NT または Windows 2000 ドメインがなく、ユーザーが各ワークステーションにローカルにログオンする場合に、マップ・ドライブ上の表構造ファイルにアクセスするには、DB2 サービス・ログオン・アカウントのユーザー名とパスワードが、そのフォルダーを共用するマシン上の有効なユーザーと同じでなければなりません。そのユーザーは、少なくとも読み取りアクセスの付与された共有フォルダーの許可リストに含まれていなければなりません。

関連資料:

- 211 ページの『Documentum ラッパーのアクセス・コントロール』

- 250 ページの『Excel ラッパーのファイル・アクセス・コントロール・モデル』

表構造ファイル・ラッパーに関する最適化のヒントと考慮事項

- システムは、ソートされていないデータ・ファイルよりもソートされたデータ・ファイルの方が、効率よく検索できます。
- ソートされたファイルの場合、照会をサブミットするときに、キー列の値または範囲を指定することによってパフォーマンスを高めることができます。
- 表構造のファイルのニックネームの統計は、SYSSTAT または SYSCAT のビューを更新することによって手操作で更新する必要があります。

関連資料:

- 132 ページの『BLAST ラッパーの最適化のヒント』

表構造ファイル・ラッパーのメッセージ

このセクションでは、表構造のファイルのラッパーを処理するときに出されるメッセージをリストして説明します。

表 85. 表構造のファイルのラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0405N	数値リテラル "<literal>" は、値が有効な値の範囲外になっているため、無効です。	データ・ファイルの列、または SQL ステートメントの述部値には、そのデータ・タイプに可能な範囲外の値が入っています。データ・ファイルを修正するか、適切なタイプに列を再定義してください。
SQL0408N	値は、その割り当てターゲットのデータ・タイプとの互換性がありません。ターゲット名は "<column_name>" です。	データ・ファイルの列に、データ・タイプに無効な文字が入っています。データ・ファイルを修正するか、適切なタイプに列を再定義してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「データ・ソース・パスが NULL」。)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「キー列取り出し障害」。)	IBM ソフトウェア・サポート担当者に連絡してください。

表 85. 表構造のファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「データ・ソースで STAT が失敗。ERRNO = <error_number>」。)	正しいディレクトリ・許可を得ていることを確認してください。ファイルが存在していることを確認してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「列情報が見つからない」。)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「サポートされていない演算子」。)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL1816N	ラッパー "<wrapper_name>" は、フェデレーテッド・データベースに定義を試みているデータ・ソース ("<type>" "") の "type" にアクセスするために使用できません。	サーバー・タイプが無効です。CREATE SERVER ステートメントにサーバー・タイプを指定してはなりません。TYPE キーワードおよび値を除去して再実行してください。
SQL1822N	データ・ソース "<server_name>" から予期しないエラー・コード "ERRNO = <error_number>" を受け取りました。関連するテキストおよびトークンは「ファイルの読み取り不可」です。	エラー番号の値を調べてください。ファイルを DB2 インスタンス所有者が読み取ることができるかどうか確認してください。次に SQL コマンドを再実行してください。
SQL1822N	データ・ソース "<server_name>" から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは「データ・ソースが標準ファイルではない」です。	データ・ソース・ファイルはディレクトリ、ソケット、または FIFO です。標準ファイルのみがデータ・ソースとしてアクセスできます。FILE_PATH オプションを変更して有効なファイルを示すようにし、SQL コマンドを再実行してください。

表 85. 表構造のファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "ERRNO = <code><error_number></code> " を受け取りました。関連するテキストおよびトークンは「ファイル・オープン・エラー」です。	ラッパーはファイルをオープンできませんでした。エラー番号を調べて、エラーが発生した原因を判別してください。データ・ソースによる問題を解決して、SQL コマンドを再実行してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは「キー列の欠落」です。	データ・ソースから取り出されたレコードはキー・フィールドが欠落しています。キー列を NULL にすることはできません。データを修正するか、ソートされていないニックネームでファイルを登録してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは「ファイルがソートされていない」です。	ファイルはキー列でソートされていませんでした。次のいずれかを実行してください。KEY_COLUMN オプションを変更して正しい列を指すようにする、データ・ファイルを再ソートする、またはニックネームをソートされていないニックネームとして登録する。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは「キーが定義サイズを超えている」です。	データ・ソースから読み取られたキー列フィールドが、ラッパー検索ルーチンが正しく機能しなくなる DB2 列定義よりも大きくなっています。データまたはニックネーム定義を修正して、ニックネームを再登録してください。
SQL1822N	データ・ソース " <code><server_name></code> " から予期しないエラー・コード "Data Error" を受け取りました。関連するテキストおよびトークンは「データ・ファイルの行が 32k を超えている」です。	データ・ファイルの行が、ラッパーによって許可されている最大長を超えています。行の長さを 32768 より大きくすることはできません。データ・ファイルの行の長さを短くしてください。
SQL1823N	サーバー " <code><server_name></code> " からデータ・タイプ " <code><data_type></code> " のためのデータ・タイプ・マッピングがありません。	サポートされていないデータ・タイプにニックネームが定義されています。サポートされているデータ・タイプだけを使ってニックネームを再定義してください。
SQL1881N	" <code><option_name></code> " は、" <code><object_name></code> " のための有効な " <code><component></code> " オプションではありません。	リストされた値は、リストされたオブジェクトに対して有効なオプションではありません。無効なオプションを削除するか変更して、SQL ステートメントを再サブミットしてください。

表 85. 表構造のファイルのラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"Nickname" オプション "COLUMN_DELIMITER" は、 "<nickname_name>" に対して "<delimiter>" に設定できません。	列区切り文字が 2 文字以上の長さになっています。オプションを 1 文字に再定義してください。次に SQL ステートメントを再実行してください。
SQL1882N	"Nickname" オプション "KEY_COLUMN" は、 "<nickname_name>" に対して "<column_name>" に設定できません。	キー列として選択された列が、このニックネームに定義されていません。このニックネームのソートされた列の 1 つになるよう KEY_COLUMN オプションを変更して、SQL コマンドを再実行してください。
SQL1882N	"Nickname" オプション "VALIDATE_DATA_FILE" は、"<nickname_name>" に対して "<option_value>" に設定 できません。	オプションの値が無効です。有効値は "Y" または "N" です。オプションを変更して、ニックネームを再登録してください。
SQL1883N	"<option_name>" は、 "<object_name>" に必須の "<component>" オプションで す。	ラッパーの必須オプションが SQL ステートメントから欠落しています。必須オプションを追加して、SQL ステートメントを再サブミットしてください。
SQL30090N	操作がアプリケーション実行 環境で無効です。理由コード = "21"。	パススルー・セッションを試行しました。表構造ファイル・ラッパーはパススルー・セッションをサポートしません。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 22 章 Teradata データ・ソースへのアクセスの構成

この章では、Teradata データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。Teradata データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、フェデレーテッド・サーバーを構成する際に実行する必要がある作業のリスト、必要な SQL ステートメントの例、およびフェデレーテッド・サーバー構成のためのチューニング情報やトラブルシューティング情報を示します。

フェデレーテッド・システムへの Teradata の追加

フェデレーテッド・サーバーへの Teradata データ・ソースの追加

Teradata データ・ソースにアクセスするようにフェデレーテッド・サーバーを構成するには、アクセスするデータ・ソースとオブジェクトに関する情報をフェデレーテッド・サーバーに提供します。

DB2 コントロール・センターまたは DB2 コマンド行を使用して、Teradata データ・ソースへアクセスするようにフェデレーテッド・サーバーを構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。
- フェデレーテッド・サーバーで Teradata クライアント・ソフトウェアがインストールおよび構成されていなければなりません。
 - Teradata バージョン V2R5 へアクセスするには、Teradata クライアントが Teradata Call-Level Interface, Version 2 (CLIV2) Release 04.07 (またはそれ以降) をサポートしていなければなりません。
 - Teradata バージョン V2R3 または V2R4 へアクセスするには、Teradata クライアントが Teradata Call-Level Interface, Version 2 (CLIV2) Release 04.06 (またはそれ以降) をサポートしていなければなりません。

手順:

Teradata データ・ソースをフェデレーテッド・サーバーに追加するには、次のようにします。

1. オプション: Teradata サーバーへの接続をテストします。
2. Teradata ライブラリーがランタイム・リンクに対応していることを検査します (AIX)。

3. Teradata ラッパーの環境変数を設定します。
4. ラッパーを登録します。
5. サーバー定義を登録します。
6. ユーザー・マッピングを作成します。
7. フェデレーテッド・サーバーから Teradata サーバーへの接続をテストします。
8. Teradata の表およびビューのニックネームを登録します。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- 61 ページの『データ・ソースの構成の近道』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 424 ページの『Teradata データ・ソースに対する構成の調整およびトラブルシューティング』
- 410 ページの『Teradata サーバーへの接続のテスト』
- 415 ページの『Teradata ラッパーの登録』
- 416 ページの『Teradata データ・ソースのサーバー定義の登録』
- 419 ページの『Teradata データ・ソースのユーザー・マッピングの作成』
- 422 ページの『Teradata 表およびビューのニックネームの登録』
- 43 ページの『フェデレーテッド・サーバーのセットアップのチェック』
- 412 ページの『Teradata 環境変数の設定』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

Teradata サーバーへの接続のテスト

Teradata サーバーへの接続のテストは、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

ラッパー、サーバー定義、またはユーザー・マッピングを作成する前に、Teradata サーバーへの接続をテストできます。接続を最初にテストして、クライアント・ソフトウェアが正しく設定されていることを確認し、CREATE WRAPPER、CREATE SERVER、CREATE USER MAPPING ステートメントを発行する際にエラーが発生するのを防いでください。

Basic Teradata Query (BTEQ) ユーティリティーを使用して SQL 照会をサブミットし、Teradata サーバーに接続できることを検査できます。BTEQ ユーティリティーの詳細については、Teradata 資料を参照してください。

前提条件:

BTEQ ユーティリティーおよび Teradata Data Connector Application Program Interface (PIOM) が、Teradata クライアント・ソフトウェアのインストール・プロセスの際にインストールされたことを確認してください。

手順:

Teradata サーバーへの接続をテストするには、次のようにします。

1. BTEQ ユーティリティー・セッションを開始し、Teradata サーバーにログオンします。
2. SQL コマンドを発行して、Teradata サーバーに正常に接続できることを検査します。例:

```
select count(*) from dbc.tables;
```

正常に接続した場合には、画面に照会出力が表示されるはずですが、例:

```
*** Query completed. One row found. One column returned.
*** Total elapsed time was 1 second.
```

```
Count(*)
```

```
-----
497
```

正常に接続できない場合には、Teradata クライアント・ソフトウェアをチェックして、フェデレーテッド・サーバー上に適切にインストールされて構成されていることをチェックしてください。

3. Teradata サーバーからログオフし、BTEQ ユーティリティー・セッションを終了します。

この一連の作業における次のタスクは、『Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)』です。

関連タスク:

- 409 ページの『フェデレーテッド・サーバーへの Teradata データ・ソースの追加』
- 411 ページの『Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)』

Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)

Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査は、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Teradata データ・ソースを AIX 上のフェデレーテッド・サーバーに追加する場合、ラッパーまたはサーバーを登録する前に、ランタイム・リンクに対応しているかどうかを検査する必要があります。

手順:

Teradata ライブラリーがランタイム・リンクに対応しているかどうかを検査するには、次のようにします。

1. libcliv2.so ファイルが存在するディレクトリーに移動します。

デフォルトでは、インストール・プロセスにより、このファイルは /usr/lib ディレクトリーに置かれます。

2. 以下の UNIX コマンドを発行します。

```
dump -H libcliv2.so | grep libtli.a
```

3. 画面に表示されるファイル名をチェックしてください。

libtli.a というファイル名が表示される場合、Teradata ライブラリーはランタイム・リンクに対応しています。

4. libtli.a というファイル名が表示されない場合は、以下の UNIX コマンドを発行してください。

```
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

このコマンドによって、Teradata ライブラリーのランタイム・リンクが可能になります。

この一連の作業における次のタスクは、『Teradata ラッパーの登録』です。

関連タスク:

- 415 ページの『Teradata ラッパーの登録』

Teradata 環境変数の設定

Teradata 環境変数の設定は、フェデレーテッド・サーバーに Teradata を追加するという、より大きなタスクの一部を成しています。

DB2 Information Integrator をインストールする際、インストール・プロセスにおいて、db2dj.ini ファイルでの Teradata 環境変数の設定が試行されます。

以下の場合には、環境変数は db2dj.ini ファイルに設定されません。

- DB2 フェデレーテッド・サーバーのセットアップ後に Teradata クライアント・ソフトウェアをインストールする場合。
- Teradata クライアント・ソフトウェアをインストールしていない場合。

Teradata の有効な環境変数は次のとおりです。

- COPERR
- COPLIB
- TERADATA_CHARSET (オプション)
- NETRACE (オプション)
- COPANOMLOG (オプション)

オプション環境変数は手動で設定する必要があります。

前提条件:

データ・ソースにアクセスするために正しくセットアップされたフェデレーテッド・サーバー。これには、データ・ソースのクライアント・ソフトウェアなどの必要なすべてのソフトウェアのインストールと構成が含まれます。

制約事項:

次のトピックを参照してください。『db2dj.ini ファイルの制約事項』

手順:

必要な環境変数を自動的に設定するには、次のようにします。

1. クライアント・ソフトウェアが DB2 フェデレーテッド・サーバーにインストールされていない場合、インストールして構成します。
2. 必要な環境変数を設定します。DB2 Information Integrator のインストールを再実行することによって、環境変数を自動的に設定できます。ランチパッドから「製品のインストール」をクリックし、ウィザードの指示に従います。

環境変数を手動で設定するには、次のようにします。

1. db2dj.ini ファイルを編集します。
 - Windows が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib\%cfg` ディレクトリーにあります。
 - UNIX が稼働するフェデレーテッド・サーバーの場合、このファイルは `sqllib/cfg` ディレクトリーにあります。

db2dj.ini ファイルには、フェデレーテッド・サーバーにインストールされている Teradata クライアント・ソフトウェアに関する構成情報が入っています。このファイルが存在しない場合、任意のテキスト・エディターを使用して、db2dj.ini という名前の新規ファイルを作成できます。db2dj.ini ファイルには、変数に完全修飾パスを指定する必要があります。そうしないと、エラーが発生します。

2. 必要に応じて以下の環境変数を設定します。

COPERR

COPERR 環境変数を、`errmsg.txt` ファイルがあるディレクトリー・パスに設定します。変数 `COPERR=teradata_lib_directory` に完全修飾パスを指定します。たとえば、次のようにします。

```
COPERR=/usr/lib
```

COPLIB

COPLIB 環境変数を、`libcliv2.so` ファイルがあるディレクトリー・パスに設定します。変数 `COPLIB=teradata_lib_directory` に完全修飾パスを指定します。たとえば、次のようにします。

```
COPLIB=/usr/lib
```

`libcliv2.so` ファイルと `errmsg.txt` ファイルは通常、同じディレクトリーにあります。

TERADATA_CHARSET

TERADATA_CHARSET 変数を設定しない場合、DB2 Information Integrator はデータベースのコード・ページに基づいてクライアント文字

セットを検出します。この変数が設定されると、DB2 Information Integrator はクライアント文字セットとして可変値を使用します。TERADATA_CHARSET 変数の値に対して妥当検査は行われませんが、正しい値に設定されないと、リモート・データ・ソースはエラーを戻します。

db2dj.ini ファイル内の TERADATA_CHARSET 環境変数を、以下の有効な文字セットのいずれかに設定します。

UNIX を実行するフェデレーテッド・サーバーの場合:

- HANGULKSC5601_2R4
- KanjiEUC_0U
- LATIN1_0A
- LATIN9_0A
- LATIN1252_0A
- SCHGB2312_1T0
- TCHBIG5_1R0
- UTF8

Windows を実行するフェデレーテッド・サーバーの場合:

- HANGULKSC5601_2R4
- KanjiSJIS_0S
- LATIN1_0A
- LATIN1252_0A
- SCHGB2312_1T0
- TCHBIG5_1R0
- UTF8

3. 環境変数がフェデレーテッド・サーバーで設定されていることを確認するには、DB2 インスタンスをリサイクルします。DB2 インスタンスをリサイクルするには、次のコマンドを発行します。

```
db2stop  
db2start
```

Teradata トレースの使用可能化および使用不能化:

Teradata NETRACE および COPANOMLOG 変数は、Teradata トレースを使用可能および使用不可にするために使用されます。Teradata サーバーに送信する SQL ステートメントのリストを保持する必要がある場合には、Teradata トレースを使用可能にできます。

Teradata トレースを使用可能にするには、db2dj.ini ファイルを編集し、それらの環境変数に以下の設定を使用します。

```
NETRACE=1  
COPANOMLOG=trace_file
```

trace_file は、トレース・データを含むファイルの完全修飾名です。こうした変数は Teradata トレース機能のみを使用可能にし、DB2 トレースには影響を及ぼしません。

Teradata トレースを使用不可にするには、db2dj.ini ファイルを編集し、NETRACE 変数と COPANOMLOG 変数の両方を除去します。

この一連の作業における次のタスクは『Teradata ラッパーの登録』です。

関連タスク:

- 422 ページの『Teradata 表およびビューのニックネームの登録』

関連資料:

- 66 ページの『db2dj.ini ファイルの制約事項』
- 66 ページの『db2dj.ini ファイルの制約事項』

Teradata ラッパーの登録

Teradata ラッパーの登録は、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Teradata データ・ソースにアクセスするためには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントを発行し、ラッパーにデフォルト名を指定します。

例:

```
CREATE WRAPPER TERADATA
```

推奨事項: TERADATA というデフォルトのラッパー名を使用してください。このデフォルト名を使用してラッパーを登録する場合、フェデレーテッド・サーバーはそのラッパー名に関連するデフォルトのライブラリー名を自動的に採用します。

ラッパー名がフェデレーテッド・データベース内の既存のラッパー名と競合する場合、デフォルトのラッパー名を、指定の名前で置き換えることができます。デフォルト名とは別の名前を使用する場合は、CREATE WRAPPER ステートメントに LIBRARY パラメーターを含める必要があります。

たとえば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバーで、tera_wrapper という名前のラッパーを登録するには、以下のステートメントを発行します。

```
CREATE WRAPPER tera_wrapper LIBRARY 'libdb2teradata.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントで指定する正しい名前については、『Teradata ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Teradata ラッパーのサーバー定義の登録』です。

関連タスク:

- 416 ページの『Teradata データ・ソースのサーバー定義の登録』

関連資料:

- 416 ページの『Teradata ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

Teradata ラッパー・ライブラリー・ファイル

以下の表は、Teradata ラッパーのディレクトリー・パスとライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で実行されている場合、ディレクトリー・パスには、libdb2teradata.a、libdb2teradataF.a、および libdb2teradataU.a のラッパー・ライブラリー・ファイルが追加されます。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 86. Teradata ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2teradata.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2teradata.sl
Solaris	/opt/IBM/db2/V8.1/lib	libdb2teradata.so
Windows	%DB2PATH%¥bin	db2teradata.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは、C:¥Program Files¥IBM¥SQLLIB です。

関連タスク:

- 415 ページの『Teradata ラッパーの登録』

Teradata データ・ソースのサーバー定義の登録

Teradata データ・ソースのサーバー定義の登録は、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

フェデレーテッド・データベース内に、アクセスしたいそれぞれの Teradata サーバーを定義する必要があります。最初に Teradata データ・ソースのノード名を見つけてから、サーバーを登録する際にそのノード名を使用しなければなりません。

手順:

Teradata データ・ソースのサーバー定義を登録するには、次のようにします。

1. ノード名を見つけます。
 - a. ホスト・ファイルを検索します。

AIX オペレーティング・システムの場合、ホスト・ファイルは `/etc/hosts` です。

Windows オペレーティング・システムの場合、ホスト・ファイルは `x:\¥WINNT¥system32¥drivers¥etc¥hosts` です。 `x:` は、¥WINNT ディレクトリが存在するドライブです。

- b. ホスト・ファイルで、リモート・サーバーの別名を検索します。

この別名は、英字ストリングで始まり、`COP n` という接尾部で終了します。値 n は、Teradata 通信プロセッサに関連付けられたアプリケーション・プロセッサの番号です。

- c. ホスト・ファイルで、この別名を含む行を検索します。
 - d. その行で最初に出現する非数値フィールドを検索します。

ホスト・ファイルの例:

```
127.0.0.1      localhost

9.22.5.77     nodexyz      nodexyzCOP1    # teradata server

9.66.111.133  rtplib05.data.xxx.com aap
9.66.111.161  rtpscm11.data.xxx.com aaprwr
9.66.111.161  rtpscm11.data.xxx.com accessm
```

この例では、**nodexyz** フィールドがノード名です。

2. CREATE SERVER ステートメントを発行します。

例:

```
CREATE SERVER server_name TYPE TERADATA VERSION 2.4 WRAPPER wrapper
OPTIONS (NODE 'node_name')
```

サーバー名を指定することが必要です。指定する名前は固有でなければなりません。

すべての Teradata サーバーに関して、TYPE パラメーターを `TERADATA` に設定しなければなりません。

Teradata ラッパーは Teradata V2R3、V2R4、および V2R5 のすべてのバージョンをサポートしています。バージョン番号は、小数点のある 2 桁で指定します。有効なバージョン番号は 2.3、2.4、2.5 です。

ラッパーの名前を指定する必要があります。指定する名前は、CREATE WRAPPER ステートメントを使用して登録した Teradata ラッパーに対応していなければなりません。

また Teradata サーバーが存在するノードの名前を指定することも必要です。このノード名には、大文字小文字の区別があります。

Teradata サーバー定義を登録する場合、必要であれば CREATE SERVER ステートメントに追加のサーバー・オプションを指定できます。

サーバー定義の登録後には、ALTER SERVER ステートメントを実行してサーバー・オプションを追加したりドロップしたりできます。

この一連の作業における次のタスクは、『Teradata データ・ソースのユーザー・マッピングの作成』です。

関連タスク:

- 419 ページの『Teradata データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 418 ページの『CREATE SERVER ステートメント - Teradata ラッパーの例』

CREATE SERVER ステートメント - Teradata ラッパーの例

ここでは、CREATE SERVER ステートメントを使用してサーバーを Teradata ラッパーに登録する方法を示すいくつかの例が提供されます。必要パラメーターのすべてを指定してサーバーを作成する方法を示すパラメーター完全指定の例と、省略可能なサーバー・オプションを指定した例とが含まれています。

完全指定の例:

次の例は、CREATE SERVER ステートメントを使用することによって、Teradata ラッパーのサーバー定義を作成する方法を示すものです。

```
| CREATE SERVER TERASERVER TYPE TERADATA  
|         VERSION 2.4 WRAPPER my_wrapper  
|         OPTIONS (NODE 'tera_node');
```

サーバー・オプション *TERASERVER* には、Teradata サーバーに割り当てる名前を指定します。TYPE *TERADATA* は、Teradata データ・ソースへのアクセスを構成していることを指定しています。VERSION 2.4 は、アクセスする Teradata サーバー・ソフトウェアのバージョン番号です。WRAPPER *my_wrapper* は、CREATE WRAPPER ステートメントによって登録した Teradata ラッパーの名前です。NODE 'tera_node' は、Teradata サーバーの存在しているノードの名前です。

サーバー・オプションの例:

次の例は、オプティマイザーのための統計データを指定した Teradata サーバー定義です。

```
| CREATE SERVER TERASERVER1 TYPE TERADATA  
|         VERSION 2.4 WRAPPER WRAPPERNAME1  
|         OPTIONS (NODE 'tera_node1', CPU_RATIO '2.0', IO_RATIO '3.0');
```

この例で、*TERASERVER1* は Teradata サーバーの名前、*WRAPPERNAME1* は CREATE WRAPPER ステートメントによって登録したラッパー名、および 'tera_node1' は Teradata サーバーの存在しているノードの名前です。CPU_RATIO と IO_RATIO のサーバー・オプションは、オプティマイザーに次の情報を提供しています。

- フェデレーテッド・サーバーの CPU リソースは、Teradata サーバーの CPU リソースの 2 倍の能力を持つ。

- フェデレーテッド・サーバーの入出力装置は、Teradata サーバーの入出力装置に比べて 3 倍の速度でデータを処理する。

関連タスク:

- 416 ページの『Teradata データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

Teradata データ・ソースのユーザー・マッピングの作成

Teradata データ・ソースのユーザー・マッピングの作成は、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

Teradata サーバーにアクセスする場合、フェデレーテッド・サーバーは、データ・ソースに対して有効なユーザー ID とパスワードを使用して、データ・ソースへの接続を確立します。それぞれのフェデレーテッド・サーバーのユーザー ID とパスワード、および対応するデータ・ソースのユーザー ID とパスワードの間の関係付け (ユーザー・マッピング) を定義しなければなりません。フェデレーテッド・システムにアクセスする、各ユーザー ID のユーザー・マッピングを作成して、Teradata データ・ソースに分散要求を送信します。

関連する CREATE SERVER ステートメントに登録した、それぞれの Teradata サーバーのユーザー・マッピングを作成する必要があります。

手順:

フェデレーテッド・ユーザー ID を Teradata サーバーのユーザー ID とパスワードにマップするには、CREATE USER MAPPING ステートメントを発行します。

例:

```
CREATE USER MAPPING FOR USERID SERVER TERASERVER
  OPTIONS (REMOTE_AUTHID 'remote_id', REMOTE_PASSWORD 'remote_password')
```

別の方法としては、DB2 コントロール・センターの「ユーザー・マッピングの作成 (Create User Mapping)」ウィンドウを使用して、ユーザー・マッピングを作成できます。

この一連の作業における次のタスクは、『フェデレーテッド・サーバーから Teradata サーバーへの接続のテスト』です。

関連タスク:

- 420 ページの『フェデレーテッド・サーバーから Teradata サーバーへの接続のテスト』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE USER MAPPING ステートメント』
- 420 ページの『CREATE USER MAPPING ステートメント - Teradata ラッパーの例』

CREATE USER MAPPING ステートメント - Teradata ラッパーの例

ここに示す例は、CREATE USER MAPPING ステートメントを使用することによって、ローカル・フェデレーテッド・ユーザー ID を、Teradata サーバーのユーザー ID およびパスワードにマップする方法を示すものです。必要パラメーターをすべて指定した完全指定の例と、CREATE USER MAPPING ステートメントで DB2 の特殊レジスター USER を使う方法を示す例とが含まれています。

完全指定の例:

次の例は、ローカル・フェデレーテッド・ユーザー ID (*MICHAEL*) を、Teradata サーバーのユーザー ID およびパスワード (*'mike'* および *'passxyz123'*) にマップする方法を示すものです。

```
CREATE USER MAPPING FOR MICHAEL SERVER TERASERVER
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123')
```

オプション *MICHAEL* は、Teradata サーバーで定義されているユーザー ID にマップするフェデレーテッド・ユーザー ID です。SERVER *TERASERVER* は、CREATE SERVER ステートメントで定義した Teradata サーバーの名前です。REMOTE_AUTHID *'mike'* は、*MICHAEL* というローカル・ユーザー ID のマップ先となる Teradata サーバー側ユーザー ID です。REMOTE_PASSWORD *'passxyz123'* は、*'mike'* の REMOTE_AUTHID 値に対応するパスワードです。

特殊レジスターの例:

以下に、特殊レジスター USER を含む CREATE USER MAPPING ステートメントの例を示します。

```
CREATE USER MAPPING FOR USER SERVER TERASERVER
  OPTIONS (REMOTE_AUTHID 'mike', REMOTE_PASSWORD 'passxyz123')
```

DB2 特殊レジスター USER を使用して、CREATE USER MAPPING ステートメントを実行する人の許可 ID を、REMOTE_AUTHID ユーザー・オプションに指定されるデータ・ソースの許可 ID にマップすることができます。

関連タスク:

- 419 ページの『Teradata データ・ソースのユーザー・マッピングの作成』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』

フェデレーテッド・サーバーから Teradata サーバーへの接続のテスト

フェデレーテッド・サーバーから Teradata サーバーへの接続のテストは、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

サーバー定義および定義したユーザー・マッピングを使用して、フェデレーテッド・サーバーから Teradata サーバーへの接続をテストできます。

手順:

接続をテストするには、次のようにします。

1. DB2 コマンド行プロセッサから、パススルー・セッションを開き、Teradata システム表に対して SQL SELECT ステートメントを発行します。

例:

```
SET PASSTHRU server_name
SELECT count(*) FROM dbc.tables
SET PASSTHRU RESET
```

SQL SELECT ステートメントがカウントを戻す場合、サーバー定義およびユーザー・マッピングは適切にセットアップされています。

2. SQL SELECT ステートメントがエラーを戻す場合には、以下のことをする必要があります。
 - Teradata サーバーを調べ、着信接続用に構成されていることを確認します。
 - ユーザー・マッピングを調べ、REMOTE_AUTHID および REMOTE_PASSWORD オプションの設定が Teradata サーバーとの接続について有効になっているかどうかを確認します。必要に応じて、ユーザー・マッピングを変更するか、別のユーザー・マッピングを作成します。
 - DB2 フェデレーテッド・サーバー上の Teradata クライアント・ソフトウェアを調べ、そのソフトウェアが正しくインストールされ、Teradata サーバーに接続するように正しく構成されていることを確認します。
 - DB2 フェデレーテッド変数の設定を調べ、Teradata サーバーにアクセスできることをチェックします。これらの変数には、システム環境変数、db2dj.ini 変数、および DB2 Profile Registry (db2set) 変数が含まれます。
 - サーバー定義を調べます。必要であれば、サーバー定義をドロップして再作成します。

パススルー・セッションを開始して Teradata オブジェクトに対する SQL ステートメントを発行する場合、ステートメントにホスト変数が含まれるなら、INTO パラメーターを伴う PREPARE ステートメントをサブミットすることはできません。

この一連の作業における次のタスクは、『Teradata 表およびビューのニックネームの登録』です。

関連タスク:

- 409 ページの『フェデレーテッド・サーバーへの Teradata データ・ソースの追加』
- 410 ページの『Teradata サーバーへの接続のテスト』
- 416 ページの『Teradata データ・ソースのサーバー定義の登録』
- 422 ページの『Teradata 表およびビューのニックネームの登録』
- 412 ページの『Teradata 環境変数の設定』

関連資料:

- ・ 「SQL リファレンス 第2巻」の『ALTER USER MAPPING ステートメント』

フェデレーテッド・サーバーでの Teradata ニックネーム

定義した各 Teradata サーバー上にある Teradata® の表およびビューのうち、アクセスしたいものごとに、ニックネームを1つずつ作成する必要があります。Teradata サーバーを照会する時には、データ・ソース・オブジェクトの名前ではなく、これらのニックネームを使用します。

フェデレーテッド・サーバーは、CREATE NICKNAME ステートメントによって割り当てたニックネームを使用することによって、Teradata データ・ソースに接続します。次にフェデレーテッド・サーバーは、データ・ソース・カタログに対して照会を実行し、データ・ソースへの接続を検査します。接続が機能しない場合、DB2® からエラー・メッセージが出されます。

フェデレーテッド・データベースは、ニックネームが付けられたオブジェクトのカタログ統計に基づいて、照会処理を最適化します。これらの統計は、データ・ソース・オブジェクトにニックネームを作成した時点で収集されます。

フェデレーテッド・データベースはデータ・ソース側にオブジェクトが存在するか検査し、次にそのデータ・ソースに基づく既存の統計データを収集します。オプティマイザーにとって役に立つ情報は、データ・ソース・カタログから読み取られ、フェデレーテッド・サーバー上のグローバル・カタログに入れます。オプティマイザーはデータ・ソースのカタログ情報の一部またはすべてを使用することがあるため、ニックネームを作成する前に、データ・ソース側で統計データを更新してください。データ・ソース側でそれらの統計データを更新するには、DB2 の **RUNSTATS** コマンドと等価のコマンドまたはユーティリティーを使用してください。

更新可能な Teradata ビューを参照するニックネームに対する SQL INSERT、UPDATE、または DELETE ステートメントは、その SQL ステートメントを Teradata データ・ソースに対して完全にプッシュダウンできるのでない限りサブミットできません。

関連タスク:

- ・ 422 ページの『Teradata 表およびビューのニックネームの登録』

関連資料:

- ・ 「コマンド・リファレンス」の『RUNSTATS コマンド』
- ・ 423 ページの『CREATE NICKNAME ステートメント - Teradata ラッパーの例』

Teradata 表およびビューのニックネームの登録

Teradata 表およびビューのニックネームの登録は、Teradata データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

定義するそれぞれの Teradata サーバーで、アクセスしたい各表およびビューごとにニックネームを登録します。

手順:

ニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

例:

```
CREATE NICKNAME TERANICKNAME FOR TERASERVER."remote_schema"."remote.table"
```

推奨事項: フェデレーテッド・データベースはニックネームが付けられたオブジェクトのカatalog統計を使用して照会処理を最適化するので、ニックネームを登録する前に Teradata データ・ソースで統計を更新してください。DB2 の **RUNSTATS** コマンドと等価のコマンドまたはユーティリティーを使用できます。

ニックネームの長さは 128 文字までです。

CREATE NICKNAME ステートメントを発行する際には、NUMERIC_STRING 列オプションを指定できます。また ALTER NICKNAME ステートメントを使用しても、この列オプションを指定できます。

関連概念:

- 422 ページの『フェデレーテッド・サーバーでの Teradata ニックネーム』

関連資料:

- 「コマンド・リファレンス」の『RUNSTATS コマンド』
- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 423 ページの『CREATE NICKNAME ステートメント - Teradata ラッパーの例』

CREATE NICKNAME ステートメント - Teradata ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して、アクセスしたい Teradata の表またはビューのニックネームを登録する方法の例が提供されます。

この例では、ニックネームが割り当てられる Teradata サーバーのリモート・オブジェクトを指定する方法が示されます。

```
CREATE NICKNAME TERASALES FOR TERASERVER."salesdata"."europe"
```

TERASALES は、Teradata の表またはビューを表すのに割り当てる固有なニックネームです。ニックネームは、2 つの部分 (スキーマと実際のニックネーム) からなる名前です。ニックネームの作成時にスキーマを省略すると、DB2 は認証 ID をスキーマとして使用してニックネームを作成します。

TERASERVER."salesdata"."europe" は、リモート・オブジェクトの ID です。それは、次の 3 つの部分で構成されます。

- TERASERVER は、CREATE SERVER ステートメントで Teradata データベース・サーバーに割り当てた名前です。
- salesdata は、表またはビューが属するリモート・スキーマの名前です。
- europe は、アクセスしたいリモートの表またはビューの名前です。

関連概念:

- 422 ページの『フェデレーテッド・サーバーでの Teradata ニックネーム』

関連タスク:

- 422 ページの『Teradata 表およびビューのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

Teradata データ・ソースに対する構成の調整およびトラブルシューティング

Teradata データ・ソースに対する構成をセットアップした後、パフォーマンスを改善し、エラーが生じる可能性を除去するために構成を変更できます。

ニックネームに対する UPDATE または DELETE 操作エラー

デフォルトでは、Teradata データ・ソース表で行が一意的に識別されることはありません。Teradata 表または Teradata ビューに関連するニックネームを更新または削除しようとする、SQL30090N、RC="21" エラーを受け取る場合があります。SQL30090N、RC="21" エラーが発生した場合、更新または削除する Teradata 表で 1 つ以上のユニーク索引を作成し、操作を再び試行してください。

Teradata アクセス・ロギングのチューニングと使用不可

Teradata 製品では、Teradata が 1 つ以上のデータベースでさまざまなユーザーの特定のセキュリティ特権をチェックする際に、ログ項目を生成するアクセス・ロギング機能が備えられています。アクセス・ロギングではかなりの量の重要なセキュリティ情報が提供されますが、この機能によりプロセッサの使用量が著しく増加し、システムのパフォーマンスが低下する可能性があります。

システムのパフォーマンスを改善することが必要な場合には、アクセス・ロギングに対して定義したチェック特権規則を評価してください。それから、END LOGGING ステートメントを定義して、不必要な規則を終了します。

最高のパフォーマンスを得るには、すべてのアクセス・ロギングをオフにします。**Teradata DBC.AccLogRules** マクロをドロップしてから、トラステッド・パラレル・アプリケーション (TPA) を強制リセットして、アクセス・ロギングを完全に停止します。

詳細については、Teradata 資料を参照してください。

libcliv2.so (AIX) のランタイム・リンクの使用可能化

djxlinkTeradata.sh ファイルを実行して libcliv2.so と呼ばれる Teradata 共有ライブラリーにリンクした場合、CREATE NICKNAME ステートメントを実行するとエラー・メッセージを受け取ることがあります。

以下は、受け取るエラー・メッセージの例です。

```
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL30081N A communication error has been detected. Communication protocol
```

```
being used: "TCP/IP". Communication API being used: "SOCKETS". Location
where the error was detected: "9.112.26.28". Communication function detecting
the error: "recv". Protocol specific error code(s): "*", "*", "0".
SQLSTATE=08001
```

エラー・メッセージを受け取る場合、/sql1lib/db2dump ディレクトリーにトラップ・ファイルがないかどうかをチェックしてください。トラップ・ファイルは、文字 t で始まり 000 という接尾部で終わる名前が付いています。例:

```
t123456.000
```

トラップ・ファイルのトレース情報で、OsCall 関数がフェデレーテッド・サーバーを停止させたことを示す OsCall 関数参照がないかどうかをチェックしてください。

以下に、トラップ・ファイル内で見つかる可能性のある OsCall 関数参照を含むトレース情報の例を示します。

```
*** Start stack traceback ***
```

```
0x239690E0 OsCall + 0x28C
0x23973FB0 mtdpassn + 0x8A4
0x239795A4 mtdp + 0x208
0x2395A928 MTDPIO + 0x28C
0x239609C4 CLICON + 0xD50
0x23962350 DBCHCL + 0xC4
```

いずれかのトラップ・ファイルに OsCall 関数参照が見つかる場合には、次の UNIX コマンドを発行します。

```
cd /usr/lib
rtl_enable libcliv2.so -F libtli.a
mv libcliv2.so libcliv2.so.old
mv libcliv2.so.new libcliv2.so
chmod a+r libcliv2.so
```

これらのコマンドにより、libcliv2.so 共有ライブラリーのランタイム・リンクが可能になります。

関連タスク:

- 409 ページの『フェデレーテッド・サーバーへの Teradata データ・ソースの追加』
- 411 ページの『Teradata ライブラリーがランタイム・リンクに対応しているかどうかの検査 (AIX)』

関連資料:

- 「コマンド・リファレンス」の『db2set - DB2 プロファイル・レジストリー・コマンド』

第 23 章 Web サービス・データ・ソースへのアクセスの構成

このセクションの情報では、Web サービス・データ・ソースをフェデレーテッド・システムに追加する方法を説明します。

Web サービス・ラッパーおよび Web サービス記述言語文書

Web サービス・プロバイダーは、Web サービス記述言語 (WSDL) 文書によって記述されています。Web サービス・ラッパーを使用して Web サービス・プロバイダーにアクセスできます。図 28 で示すように、Web サービス・プロバイダーはサービスをインプリメントし、そのインターフェースを UDDI などのサービス・ブローカーへ公開します。その結果、サービス要求元はサービス・ブローカーを使用して Web サービスを検索することができます。リクエスターがサービスを検出すると、リクエスターは Web サービスを使用できるようそのサービス・プロバイダーにバインドします。リクエスターは、リクエスターとプロバイダー間で SOAP (Simple Object Access Protocol) メッセージを交換することによりサービスを呼び出します。

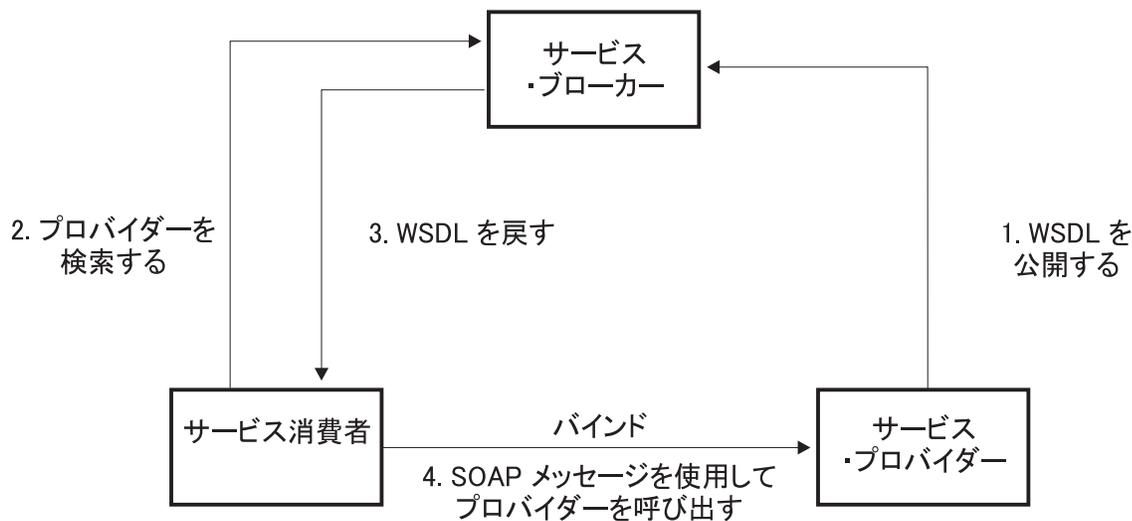


図 28. Web サービス: サービス指向アーキテクチャー

SOAP 仕様は XML に基づいたメッセージのレイアウトを定義します。SOAP メッセージは SOAP エンベロープに含まれています。エンベロープはオプションの SOAP ヘッダーと必須の SOAP 本体から成っています。SOAP ヘッダーには暗号化情報あるいは認証情報といったメッセージに関する情報が含まれます。SOAP 本体にメッセージが含まれます。SOAP 仕様は、SOAP エンコードと呼ばれるプログラム言語バインディングのデフォルト・エンコードも定義します。

Web サービスへのキーは WSDL 文書です。WSDL 文書は、Web サービスをそれが送受信するメッセージにおいて説明する XML 文書です。メッセージは、通常は XML スキーマであるタイプ・システムを使用して記述されます。Web サービス操作はメッセージ交換パターンを 1 つ以上のメッセージと関連付けます。メッセージ

交換パターンは、送信または受信されたメッセージのシーケンスとカーディナリティー、またメッセージの論理的な送信先または送信元を識別します。インターフェースは、トランスポートまたはワイヤー・フォーマットへのコミットメントなしに操作をグループ化します。WSDL バインディングは 1 つ以上のインターフェースに対して、トランスポートおよびワイヤー・フォーマット詳細を指定します。エンドポイントはネットワーク・アドレスをバインディングと関連付けます。サービスは、共通のインターフェースをインプリメントするエンドポイントをグループ化します。メッセージは、文書指向の情報または、リモート・プロシージャー・コール (RPC) としても知られるプロセス指向の情報を含むことができます。WSDL 文書は、1 つ以上の Web サービスを含むことが可能です。

429 ページの図 29 の例は株価見積価格を供給する簡単なサービスの WSDL 定義を示します。Web サービスは、GetLastTradePrice という名前の単一の操作をサポートします。このサービスは、HTTP を通して SOAP 1.1 プロトコルでアクセスできます。要求は、ストリング・データ・タイプであるチッカー・シンボルを入力として読み取り、浮動データ・タイプである価格を戻します。ストリングおよび浮動データ・タイプは XML スキーマ規格の事前定義タイプです。Web サービスもデータ・タイプを定義し、ユーザー定義のデータ・タイプを使用することができます。事前定義およびユーザー定義の XML データ・タイプはニックネームの列にマップします。完全な例と WSDL 仕様は、W3C Web サイトにあります。

```

<?xml version="1.0"?>
<definitions name="StockQuote"
...

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>

<message name="GetLastTradePriceInput">
...
</message>

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

<binding name="StockQuoteSoapBinding"
  type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
</definitions>

```

図 29. WSDL 文書の例

Web サービス・ラッパーは、HTTP トランスポートと SOAP バインディングのあるポート・タイプの操作を使用します。操作中の入力メッセージおよび関連したタ

イブまたはエレメントがニックネーム中の列になります。操作中の出力メッセージはニックネーム階層へ抽出されます。WSDL 文書中で、それぞれの操作ごとに別々のニックネームの階層を作成できます。

図 30 では、操作名 GETTEMP のポート・タイプを含む WSDL 文書を使用します。この Web サービスでは、入力データとして郵便番号を入力し、その郵便番号の地域の温度を受信します。

```
<?xml version="1.0"?>
<definitions name="TemperatureService" targetNamespace=http://www.xmethods.net/
  sd/TemperatureService.wsdl"
  xmlns:tns="http://www.xmethods.net/sd/TemperatureService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="getTempRequest">
    <part name="zipcode" type="xsd:string"/>
  </message>
  <message name="getTempResponse">
    <part name="return" type="xsd:float"/>
  </message>
  <portType name="TemperaturePortType">
    <operation name="getTemp">
      <input message="tns:getTempRequest"/>
      <output message="tns:getTempResponse"/>
    </operation>
  </portType>
  <binding name="TemperatureBinding" type="tns:TemperaturePortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getTemp">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded" namespace="urn:xmethods-Temperature"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="TemperatureService">
    <documentation>
      Returns current temperature in a given U.S. zipcode
    </documentation>
    <port name="TemperaturePort" binding="tns:TemperatureBinding">
      <soap:address
        location="http://services.xmethods.net:80/soap/servlet/rpcrouter" />
    </port>
  </service>
</definitions>
```

図 30. GETTEMP Web サービス

入力値は、zipcode 列で記述されています。出力値は、return 列で記述されています。WSDL 文書では、これらの列はメッセージ・エレメントの中で識別されます。メッセージ・エレメントは、Web サービス・プロバイダーと Web サービス消費者との間で送信されるデータの論理定義を表します。メッセージ・エレメント中の情報にさらに説明が必要である場合、WSDL 文書にもタイプ・エレメントが含まれます。

す。タイプ・エレメントは、XML スキーマ仕様に基づいた事前定義タイプ、またはユーザーによって定義されたタイプを意味します。

図 31 では、DB2[®] コントロール・センターの Discover ツールが WSDL 文書から作成するニックネームを示します。zipcode 列は、ニックネーム TEMPLATE 構文があるため必須入力列です。

```
CREATE NICKNAME GETTEMP (  
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),  
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')  
)  
FOR SERVER "EHPWSSERV"  
OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',  
  SOAPACTION ' ',  
  TEMPLATE '<soapenv:Envelope>  
    <soapenv:Body>  
      <ns2:getTemp>  
        <zipcode>&zipcode[1,1]</zipcode>  
      </ns2:getTemp>  
    </soapenv:Body>  
  </soapenv:Envelope>',  
  XPATH '/soapenv:Envelope/soapenv:Body/*',  
  NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",  
    ns2="urn:xmethods-Temperature" ,  
    soapenv="http://schemas.xmlsoap.org/soap/envelope/" );
```

図 31. GETTEMP ニックネーム

Web サービス・ラッパーのニックネーム・オプション URL および SOAPACTION は、ニックネーム作成時に指定したエンドポイントまたはアドレスをオーバーライドする機能を提供します。照会に URLCOLUMN または SOAPACTIONCOLUMN を使用可能にした列を使用する場合、同じニックネームにダイナミック・アドレスを使用できます。ニックネーム作成時に、ニックネーム・オプション URL および SOAPACTION を定義し、列オプションで URLCOLUMN および SOAPACTIONCOLUMN を使用可能にした場合、Web サービス・ラッパーの遅延バインディング機能を使用していることとなります。SOAPACTION ニックネーム・オプションの値が HTTP ヘッダー中の属性となります。URL ニックネーム・オプションの値は、要求の送信先である HTTP URL です。

URL および SOAPACTION ニックネーム・オプションは、ダイナミック・ニックネーム関連を提供します。これらのダイナミック・アドレスは、いくつかの会社が 1 つの Web サービス・ポート・タイプをインプリメントする場合に役立ちます。Web サービス・ラッパーには、WSDL 文書間の違いが異なる URL および SOAPACTIONS のみであることが必要です。遅延バインディング機能を使用して、異なる会社が使用する異なるサービス・エンドポイントに対して同じニックネームを作成、使用することができます。URL および SOAPACTION 値は、WSDL 文書から取られています。

以下の例では、URLCOLUMN および SOAPACTIONCOLUMN 列オプションを使用する方法を示します。

```

CREATE NICKNAME GetPartQuote(
  partnumber INTEGER OPTIONS (TEMPLATE'&column'),
  price FLOAT OPTIONS (XPATH './price'),
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),
FOR SERVER myServer
OPTIONS (
  ...
  SOAPACTION 'http://example.com/GetPartPrice' ,
  URL 'http://mycompany.com:9080/GetPartPrice',
  ...
)

```

図 32. *GetPartQuote* ニックネーム

以下の例では、URLCOLUMN 列オプションおよび SOAPACTIONCOLUMN 列オプションが使用可能で定義された列、URLCOL および SOAPACTIONCOL を使用します。

```

SELECT * FROM supplier_endpoints p,
       GetPartQuote q
WHERE partnumber=1234 AND
       p.url=q.urlcol AND
       p.soapaction=q.soapactioncol;

```

SQL アプリケーションは、ニックネーム作成時に特定のエンドポイントを定義する代わりに、使用するエンドポイントの選択を照会が実行されるまで延期することができます。

Web サービス・ラッパーは、大量の WSDL 文書データをフラグメントに分離して、使用されるメモリの合計を縮小することができます。Web サービス・ニックネームの作成時に、DB2 コントロール・センターの「プロパティ」ウィンドウの「設定」ページで **STREAMING** オプションを指定してください。Web サービス・ラッパーは、その結果得られる XML データのストリームを処理し、それから照会フラグメントによって要求されている情報を抽出します。Web サービス・ラッパーは、フラグメントを一度に 1 つずつ解析します。**STREAMING** オプションを使用するのは、大規模な XML 文書を解析する場合だけにしてください。

関連概念:

- 「*IBM DB2 Information Integrator アプリケーション開発者向けガイド*」の『DADX ファイルからの WSDL』
- 「*IBM DB2 Information Integrator アプリケーション開発者向けガイド*」の『Web サービス記述言語』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 434 ページの『Web サービス・ラッパーの登録』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』

Web サービスのフェデレーテッド・システムへの追加

フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加

フェデレーテッド・サーバーが Web サービス・データ・ソースにアクセスするように構成するには、Web サービス記述言語 (WSDL) 文書などの、アクセスしたいデータ・ソースとオブジェクトの情報をサーバーに提供する必要があります。

フェデレーテッド・サーバーの Web サービス・データ・ソースへのアクセスは、DB2 コントロール・センターまたは DB2 コマンド行を使用して構成できます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成に必要な手順を示すウィザードが含まれています。

前提条件:

- ・ DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- ・ フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

フェデレーテッド・サーバーに Web サービス・データ・ソースを追加するには、次のようにします。

1. ラッパーを登録します。
2. サーバー定義を登録します。
3. オプション: ユーザー・マッピングを作成します。
4. Web サービス・データ・ソースのニックネームを登録します。
5. オプション: Web サービス・ニックネームのフェデレーテッド・ビューを作成します。

関連概念:

- ・ 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』
- ・ 459 ページの『Web サービス・ラッパーのセキュリティー』
- ・ 459 ページの『Web サービス・ラッパーのセキュリティー』
- ・ 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- ・ 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール後の Oracle genclntsh スクリプトの編集および libclntsh ファイルの作成 (HP-UX, Linux, Solaris)』
- ・ 434 ページの『Web サービス・ラッパーの登録』
- ・ 435 ページの『Web サービス・データ・ソースのサーバー定義の登録』
- ・ 436 ページの『Web サービス・データ・ソースのニックネームの登録』

- 445 ページの『Web サービス・ニックネームのフェデレーテッド・ビューの作成』
- 57 ページの『フェデレーテッド・データベースの作成』
- 434 ページの『Web サービス・ラッパーの登録』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 460 ページの『Web サービス・ラッパーのメッセージ』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

Web サービス・ラッパーの登録

Web サービス・ラッパーの登録は、フェデレーテッド・サーバーに Web サービス・データ・ソースを追加するという、より大きな作業の一部を成しています。

Web サービス・データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、CREATE WRAPPER ステートメントをラッパーの名前およびラッパー・ライブラリー・ファイル名と共に発行します。

たとえば、Windows オペレーティング・システムを使用するフェデレーテッド・サーバーで websr_wrapper という名前のラッパーを登録するには、次のステートメントを発行します。

```
CREATE WRAPPER websr_wrapper LIBRARY 'db2ws.d11';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ラッパー・ステートメントで指定する正しい名前については、『Web サービス・ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『Web サービス・ラッパーのサーバー定義の登録』です。

関連概念:

- 459 ページの『Web サービス・ラッパーのセキュリティー』
- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 435 ページの『Web サービス・データ・ソースのサーバー定義の登録』

関連資料:

- 435 ページの『Web サービス・ラッパー・ライブラリー・ファイル』

Web サービス・ラッパー・ライブラリー・ファイル

次の表には、Web サービス・ラッパーのディレクトリー・パスおよびライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは libdb2ws.a、libdb2wsF.a、および libdb2wsU.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 87. Web サービス・ラッパー・ライブラリーの位置およびファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2ws.a
Windows	%DB2PATH%\bin	db2ws.dll

%DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

関連タスク:

- 434 ページの『Web サービス・ラッパーの登録』

Web サービス・データ・ソースのサーバー定義の登録

Web サービス・データ・ソースのサーバー定義の登録は、フェデレーテッド・システムに Web サービスを追加するという、より大きな作業の一部を成しています。

ラッパーを登録した後、対応するサーバーを登録する必要があります。アクセスする Web サービスそれぞれに対して、サーバー定義を登録しなければなりません。

サーバー定義は、DB2 コマンド行または DB2 コントロール・センターから登録できます。

手順:

Web サービス・ラッパーのサーバー定義をフェデレーテッド・システムに登録するには、CREATE SERVER ステートメントを発行します。

たとえば、Windows で `ws_server` という名前の Web サービス・サーバー定義を登録するには、次のステートメントを発行します。

```
CREATE SERVER ws_server WRAPPER websr_wrapper;
```

この一連の作業における次の作業は、『Web サービス・データ・ソースのニックネームの登録』です。

関連タスク:

- 434 ページの『Web サービス・ラッパーの登録』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

Web サービス・データ・ソースのニックネームの登録

Web サービス・データ・ソースのニックネームの登録

Web サービス・データ・ソースのニックネームの登録は、フェデレーテッド・システムに Web サービスを追加するという、より大きな作業の一部を成しています。

それぞれの Web サービス操作ごとに 1 つのニックネーム階層を作成します。Web サービス操作は、Web サービス記述言語 (WSDL) 文書の中で定義されています。親ニックネームには少なくとも 1 つの子ニックネームが含まれます。子ニックネームは、親ニックネームの要素内でネストされている要素に対応します。

ニックネームは、DB2 コマンド行または DB2 コントロール・センターから作成できます。

DB2 コントロール・センターでは、Discovery ツールを使用してニックネームを速く作成できます。Discover ツールへの入力は、WSDL 文書の位置の URL です。Discover ツールは、WSDL 文書の処理の結果としてニックネームを作成します。WSDL 文書には、WSDL ファイルまたは WSDL ファイルにインポートされた外部 XML スキーマ・ファイルに組み込まれたスキーマ定義が含まれていることがあります。これらのスキーマ定義は、URL アドレスの使用によってインポートされます。

前提条件:

通信したい Web サービスを説明する有効な WSDL 文書へアクセスできる必要があります。

制約事項:

- 要求/応答の操作のみがサポートされています。
- HTTP トランスポートの SOAP バインディングのみがサポートされるバインディングです。

- SOAPACTIONCOLUMN、URLCOLUMN、PRIMARY_KEY または FOREIGN_KEY オプションの特殊な列以外は、それぞれの列にTEMPLATE オプションまたは XPATH オプションを使用しなければなりません。

手順:

DB2 コマンド行から Web サービス・データ・ソースのニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

たとえば、GETTEMP という名前の Web サービスのために Windows でニックネームを登録するには、以下のステートメントを発行します。

```
CREATE NICKNAME GETTEMP (
  ZIPCODE VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  RETURN VARCHAR (48) OPTIONS(XPATH './return/text()')
)
FOR SERVER "EHPWSSERV"
OPTIONS(URL 'http://services.xmethods.net:80/soap/servlet/rpcrouter',
  SOAPACTION ' ',
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <ns2:getTemp>
        <zipcode>&zipcode[1,1]</zipcode>
      </ns2:getTemp>
    </soapenv:Body>
  </soapenv:Envelope>',
  XPATH '/soapenv:Envelope/soapenv:Body/*' ,
  NAMESPACES ' ns1="http://www.xmethods.net/sd/TemperatureService.wsdl",
    ns2="urn:xmethods-Temperature" ,
    soapenv="http://schemas.xmlsoap.org/soap/envelope/"');
```

DB2 コントロール・センターが生成するデータ定義言語 (DDL) は、すべての入力エレメントを、ニックネーム階層中のルート・ニックネームの列にマップします。作成されるニックネームは、WSDL 文書から取られています。

DB2 コントロール・センターから Web サービス・データ・ソースのニックネームを登録するには、次のようにします。

1. 「フェデレーテッド・データベース・オブジェクト」フォルダーを展開します。
2. ニックネームを登録するラッパー・フォルダーを展開します。
3. 「サーバー定義 (Server Definitions)」フォルダーを展開します。
4. ニックネームを登録するサーバー・フォルダーを展開します。
5. 「ニックネーム (Nicknames)」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。
6. 「ニックネームの作成 (Create Nicknames)」ウィンドウで「発見 (Discover)」をクリックして、データ・ソースでオブジェクトを選択するのに役立つ検索条件を定義します。
7. DB2 DB2 Information Integrator ユーザーにアクセスさせたい Web サービスの定義を含む WSDL 文書を指定します。

WSDL 文書はローカル文書としたり、その位置を URL を使用して指定することもできます。

8. 「OK」をクリックして、選択した WSDL 文書に応じてニックネームを作成します。

DB2 コントロール・センターは、複数のニックネーム作成 DDL ステートメントに対して、適切な親子関係定義で WSDL 文書を抽出します。作成されるニックネームは、WSDL 文書で定義された Web サービス階層を表します。

オプション: この一連の作業における次のタスクは、『Web サービス・ニックネームのフェデレーテッド・ビューの作成』です。

関連概念:

- 459 ページの『Web サービス・ラッパーのセキュリティー』

関連タスク:

- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 434 ページの『Web サービス・ラッパーの登録』

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 460 ページの『Web サービス・ラッパーのメッセージ』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ニックネームおよび列レベルでの TEMPLATE オプション

このトピックは、WebSphere® Business Integration ラッパーおよび Web サービス・ラッパーに適用されます。

WebSphere® Business Integration ラッパーと Web サービス・ラッパーは、WebSphere Business Integration アダプターと Web サービス環境に必要な XML 文書を作成します。これらのラッパーにはニックネーム作成時にニックネーム・レベルおよび列レベルのテンプレート・フラグメント (つまり、CREATE NICKNAME ステートメントの TEMPLATE オプション) が必要です。ラッパーは、照会計画中および照会実行段階で、この情報を使用します。

Web サービス・ラッパー

Web サービス・ラッパーの場合、必須およびオプションの属性は、WSDL 文書内の定義、および列の導出方法によって異なります。列は、エレメントまたはエレメントの属性のいずれから導出されます。

- 列がエレメントから導出されている場合、列がオプションかどうかは minOccurs 値によって判別されます。
- minOccurs の値が 0 である場合、列はオプションです。
- minOccurs の値が 1 である場合、列は必須です。
- 列がエレメントの属性から導出されている場合、列がオプションかどうかは属性に使用されている値により判別されます。
- 属性に値 use=optional が含まれている場合、列はオプションです。

- 属性に値 use=required が含まれている場合、列は必須です。

次の例は、列に関連付けられているスキーマ定義内の属性を示しています。

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="tns:ZooName"/>
    <xsd:element ref="tns:Count"/>
    <xsd:element ref="tns:LastModified"/>
    <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:complexType>
```

WebSphere Business Integration ラッパー

WebSphere Business Integration ラッパーの場合、列が必須かオプションかは、アプリケーションおよび関連付けられているアダプターによって異なります。列に適したテンプレート・オプション値を指定することによって、必須およびオプションの入力列を識別する必要があります。DB2[®] コントロール・センターを使用してニックネームを作成する前に、XMLスキーマ定義ファイルを変更して、必須およびオプション入力列にフラグを立てなければなりません。

SAP BAPI

IBM[®] DB2 コントロール・センターは、XML スキーマ定義 (XSD) ファイル内にあるビジネス・オブジェクト定義を表す固有のフラグの値に基づいて、必須およびオプション入力列を判別します。

任意のレベルのビジネス・オブジェクト階層 (親または子ビジネス・オブジェクト) でエレメントの注釈セクションで、appSpecificInfo 値に I 接頭部があれば、ビジネス・オブジェクト定義のマップ先である SAP BAPI のインポート・パラメーターを表します。E 接頭部は、SAP BAPI のエクスポート・パラメーターを表します。一部のエレメントは、BAPI のインポートおよびエクスポートの両方のパラメーターとなることができます。次の例は、インポートおよびエクスポートの両方のパラメーターであるエレメントを示しています。

```
<bx:appSpecificInfo>ICOMPANYCODE:ECOMPANYCODE</bx:appSpecificInfo>
```

SAP ビジネス・オブジェクト・リポジトリから抽出された情報に基づいて、接頭部は WebSphere Business Integration Object Discovery Agent ツールにより自動的に生成されます。

インポート・パラメーター (appSpecificInfo 値に I 接頭部が指定されている) を表すエレメントに属性 minOccurs=1 が指定されている場合、DB2 コントロール・センターはそのエレメントを必須入力パラメーターとして識別し、エレメントを必須入力列としてニックネーム定義にフラグを立てます。WebSphere Business Integration Object Discovery Agent ツールは、SAP BAPI の必須入力パラメーターに対して自動的に minOccurs の値 1 を設定することはありません。ユーザー自身が SAP Business Object Repository を参照して、アクセスする BAPI 用の必須入力パラメーターすべてを判別する必要があります。次いで、XML スキーマ・ファイル内の対応するエレメントについて、手動で属性を minOccurs=1 に設定して編集する必要があります。入力パラメーターの minOccurs 属性値がデフォルト値である 0 のま

までであると、DB2 コントロール・センターは生成されるニックネーム階層に列をオプション入力列として指定します。

次の例は、オプション入力列を示しています。

```
<xsd:element name="Company_code" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICOMPANYCODE:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

次の例は、必須入力列を示しています。

```
<xsd:element name="Company_id" minOccurs="1">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>ICOMPANYID:</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="true" isKey="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="4" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

SAP ビジネス・アプリケーション用の必須およびオプション入力列は、次の表に示されている構文によって指定されます。

表 88. SAP 入力列情報のフラグ設定スキーマ

SAP XSD ファイルで使用されるフラグ	必要入力列	ニックネーム・テンプレート内の列参照
接頭部 = 'I' および minOccurs=1 が指定された、階層の任意の場所にあるエレメント	はい	&columnname[1,1]
接頭部 = 'I' および minOccurs=0 が指定された、階層の任意の場所にあるエレメント	いいえ	&columnname[0,1]

Siebel および PeopleSoft

DB2 コントロール・センターは、エレメントの注釈の attributeInfo セクションにおける isRequired フラグの有無およびその値に基づいて、必須およびオプション入力列を判別します。isRequired フラグがない場合、その列は入力列ではありません。WebSphere Business Integration Object Discovery Agent ツールは、XSD ファイルにこれらのフラグを自動的に生成することはありません。ユーザー自身が必須およびオプション入力列を識別し、DB2

コントロール・センターを使用してニックネーム DDL を生成する前に、XSD ファイル内でそれらの列について適切にフラグを立てる必要があります。

次の例は、Siebel または PeopleSoft ビジネス・オブジェクト定義用の XSD ファイル内での必須入力列およびオプション入力列のフラグを示しています。

```
<xsd:element name="sieb_ssa_Contact_Contact">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boDefinition version="1.0.0">
        <bx:appSpecificInfo>ON=Contact;CN=Contact</bx:appSpecificInfo>
      </bx:boDefinition>
    </xsd:appinfo>
  </xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Id" minOccurs="0">
  <xsd:annotation>
    <xsd:appinfo>
      <bx:boAttribute>
        <bx:appSpecificInfo>FN=Id</bx:appSpecificInfo>
        <bx:attributeInfo isForeignKey="false"
          isKey="true" isRequired="true" />
      </bx:boAttribute>
    </xsd:appinfo>
  </xsd:annotation>
  ...
</xsd:element>
...

```

図 33. Siebel ビジネス・オブジェクト定義の一部分 (1/2)

```

...
<xsd:element name="FirstName" minOccurs="1">
<xsd:annotation>
  <xsd:appinfo>
    <bx:boAttribute>
      <bx:appSpecificInfo>FN=First Name</bx:appSpecificInfo>
      <bx:attributeInfo isForeignKey="false" isKey="false"
        isRequired="false" />
    </bx:boAttribute>
  </xsd:appinfo>
</xsd:annotation>
<xsd:simpleType>
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="50" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="LastName" minOccurs="1">
<xsd:annotation>
  <xsd:appinfo>
    <bx:boAttribute>
      <bx:appSpecificInfo>FN=Last Name</bx:appSpecificInfo>
      <bx:attributeInfo isForeignKey="false" isKey="false"
        isRequired="false" />
    </bx:boAttribute>
  </xsd:appinfo>
</xsd:annotation>
...

```

図 33. Siebel ビジネス・オブジェクト定義の一部分 (2/2)

Siebel および PeopleSoft ビジネス・アプリケーション用の必須およびオプション入力列は、次の表に示されている構文によって指定されます。

表 89. Siebel および PeopleSoft 入力列情報のフラグ設定スキーマ

Siebel および PeopleSoft XSD ファイルで使用されるフラグ	必要入力列	ニックネーム・テンプレート内の列参照
isRequired="true"	はい	&columnname[1,1]
isRequired="false"	いいえ	&columnname[0,1]

次の例は、Siebel ビジネス・オブジェクト定義の一部分 の図に示されている XSD ファイルに基づいて、DB2 コントロール・センターが作成する DDL を示しています。この図の XSD ファイルでは、isRequired 属性の値が false に指定されて組み込まれています。

```

CREATE NICKNAME sieb_ssa_Contact_Contact_NN(
  Id VARCHAR(15)  OPTIONS(XPATH './ns1:Id/text()'),
    TEMPLATE '<ns1:Id>&column</ns1:Id>'),
  FirstName VARCHAR(50)  OPTIONS(XPATH './ns1:FirstName/text()'),
    TEMPLATE '<ns1:FirstName>&column</ns1:FirstName>'),
  LastName VARCHAR(50)  OPTIONS(XPATH './ns1:LastName/text()'),
    TEMPLATE '<ns1:LastName>&column</ns1:LastName>'),
  AccountId VARCHAR(255)  OPTIONS(XPATH './ns1:AccountId/text()'),
  PrimaryAccountName VARCHAR(100)
    OPTIONS(XPATH './ns1:PrimaryAccountName/text()'),
  PrimaryPostalCode VARCHAR(30)
    OPTIONS(XPATH './ns1:PrimaryPostalCode/text()'),

```

```

PrimaryStreetAddress VARCHAR(200)
    OPTIONS(XPATH './ns1:PrimaryStreetAddress/text()'),
SalesRep VARCHAR(255) OPTIONS(XPATH './ns1:SalesRep/text()'),
State VARCHAR(255) OPTIONS(XPATH './ns1:State/text()')
FOR SERVER siebel_server
OPTIONS(XPATH '//ns1:sieb_ssa_contact_contact',
TEMPLATE '<ns1:sieb_ssa_contact_contact>
    &Id[1,1] &FirstName[0,1] &LastName[0,1]
</ns1:sieb_ssa_contact_contact>',
BUSOBJ_NAME 'sieb_ssa_contact_contact',
NAMESPACES 'ns1="http://www.ibm.com/websphere/
crossworlds/2002/BOSchema/sieb_ssa_contact_contact"');

```

関連概念:

- 135 ページの『WebSphere Business Integration ラッパー』
- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 460 ページの『Web サービス・ラッパーのメッセージ』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ニックネームと XPATH 式

このトピックは、WebSphere® Business Integration ラッパーおよび Web サービス・ラッパーに適用されます。

ニックネームは、XML 文書データのツリー構造に対応しています。親ニックネームと子ニックネームは、データ・ツリー構造のルート構造とネスト・エレメントに対応します。それら親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。

各ニックネームは、出力値を表す XPath 式によって定義されます。WebSphere Business Integration ラッパーおよび Web サービス・ラッパーは、XPath 式を使用することによって、XML 文書のデータと、リレーショナル表の行との間の対応を確立します。それらの XPath 式は、XML 文書内の値を特定し、それらの値が各行の列にどう対応するかを決定します。WebSphere Business Integration ラッパーおよび Web サービス・ラッパーは、XML 文書データのみを読み取ります。ラッパーは、

データを更新しません。 XPATH オプションには、SOAP エンベロープおよび SOAP ボディ・タグを通して SOAP メッセージを検索するための情報が含まれています。 getQuote メッセージは、SOAP エンベロープおよびボディ・エレメントに含まれています。

NICKNAME オプション XPATH 式は、出力エレメント中の繰り返しタグを指します。 XPath 式は、ニックネーム中にいくつの行、またはどの行が含まれるかを決定します。列オプション XPATH 式は、NICKNAME XPATH 式に関係します。列オプション XPATH は行中の値を示します。子ニックネーム中の NICKNAME オプション XPATH は、親ニックネーム中の NICKNAME オプション XPATH 式に関係します。

ニックネームを作成する際、そのニックネームと XML 文書との間の関連付けを指定するためのオプションを選択します。 WebSphere Business Integration ラッパー用に作成されたニックネームは、XML スキーマ定義 (XSD) 文書と関連しています。 Web サービス・ラッパー用に作成されたニックネームは、Web サービス記述言語 (WSDL) 文書と関連しています。

関連概念:

- 463 ページの『XML とは?』
- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』
- 471 ページの『ニックネームと XML 文書間のデータ関連』

関連タスク:

- 466 ページの『フェデレーテッド・システムへの XML の追加』
- 474 ページの『XML データ・ソースのニックネームの登録』
- 479 ページの『非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)』
- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 146 ページの『ビジネス・アプリケーション・データ・ソースのニックネームの登録』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

Web サービス・ニックネームのフェデレーテッド・ビューの作成

Web サービス・ニックネームのフェデレーテッド・ビューの作成は、フェデレーテッド・システムに Web サービス・データ・ソースを追加するという、より大きな作業の一部を成しています。

Web サービス文書を説明するニックネームの階層のフェデレーテッド・ビューを定義できます。フェデレーテッド・ビューを定義すると、Web サービス・ニックネーム階層の断片を結合する照会が適切に実行されるようになります。

手順:

Web サービス・ニックネームのフェデレーテッド・ビューを定義するには次のようにします。

- 1 つ以上の Web サービス・ニックネームのビューを定義します。Web サービスの 1 つの操作に関連するすべてのニックネームを結合したい場合、それらのニックネームすべてを含むビューを定義する必要があります。
- ビューの WHERE 文節では、PRIMARY_KEY および FOREIGN_KEY 列オプションによって関連するすべての列に対して、結合述部を使用します。

以下の例では、主キーは、ニックネーム zooport_getzooreport_report_nn 中の列 ooport_getzooreport_pk にあります。外部キーは、ニックネーム zooport_getzooreport_report_report_nn 中の列 ooport_getzooreport_fkey にあります。

```
CREATE VIEW zooreport
(zooid, zooname, number_of_zookeeper,
lastmodified, zookeeper_id, zookeeper_name,
fingers_left, animal_name, animal_species, animal_lot)
AS ( SELECT zooid, report_zooname,
report_count, report_lastmodified,
zookeeper_id, zk.report_name, report_numberfingersleft,
a.report_name, report_species,
report_lot
FROM zooport_getzooreport_nn ,
zooport_getzooreport_report_nn as zk,
zooport_getzooreport_report_report_nn as a
WHERE zk.ooport_getzooreport_pkey=a.ooport_getzooreport_fkey
AND zooport_getzooreport_pkey=zk.ooport_getzooreport_fkey);
```

以下の SELECT ステートメントで、すべてのニックネームからの情報を得ることができます。

```
SELECT * FROM zooreport WHERE zooid='1';
```

一連の作業はこれで終わりです。

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

CREATE NICKNAME ステートメント - Web サービス・ラッパーの例

Web サービスへのアクセスのためにニックネームを作成する場合、Web サービス操作の入力メッセージ中のそれぞれの値ごとに入力列を、またWeb サービス操作の出力メッセージ中のそれぞれの値ごとに出力列を作成します。その入出力列定義は、ニックネーム列オプション定義でコントロールします。

TEMPLATE 列オプションは、列が入力列であることを指定します。 XPATH 列オプションは、列が出力列であることを指定します。 TEMPLATE ニックネーム・オプションに大括弧で囲まれた表記 ([1,1]) が含まれる場合、列は必須入力列です。 TEMPLATE ニックネーム・オプションに大括弧で囲まれた表記 ([0,1]) が含まれる場合、列はオプションの入力列です。

NAMESPACES ニックネーム・オプションは、入出力 XML 文書のエレメントに使用されるニックネームを解決するためにフェデレーテッド・システムが使用する、名前値のペアのコンマで区切られたリストです。 NAMESPACES はメッセージ要求に使用され、TEMPLATE ニックネーム・オプションに使用される接頭部が定義されます。 NAMESPACES ニックネーム・オプションは、WSDL または XML スキーマで定義された名前・スペース URI がある XPath 式で使用されている接頭部を解決するために使用されます。XPath 式は、Web サービスから戻される XML 文書で適用されます。

例 1: 必須入力列

以下の例では、getQuote という名前の Web サービスを使用するニックネームを示します。この Web サービスは株式チャッカー・シンボルを読み取り、取引価格を戻します。以下の DDL は、DB2 コントロール・センターの Discover ツールによって作成されます。

```
CREATE NICKNAME "stockquote.stockquoteport_getquote_nn" (  
  symbol VARCHAR (48) OPTIONS(TEMPLATE '&column'),  
  result VARCHAR (48) OPTIONS(XPATH './Result/text()'))  
FOR SERVER "xmethods_server" OPTIONS(  
  URL 'http://66.28.98.121:9090/soap' ,  
  SOAPACTION 'urn:xmethods-delayed-quotes#getQuote' ,  
  TEMPLATE '<soapenv:Envelope>  
    <soapenv:Body>  
      <ns2:getQuote>  
        <symbol>&symbol [1,1]</symbol>  
      </ns2:getQuote>  
    </soapenv:Body>  
  </soapenv:Envelope>',  
  XPATH '/soapenv:Envelope/soapenv:Body/*' ,  
  NAMESPACES 'ns2="urn:xmethods-delayed-quotes" ,  
    ns1="http://www.themindelectric.com/wsd1/  
    net.xmethods.services.stockquote.StockQuote/" ,  
    soapenv="http://schemas.xmlsoap.org/soap/envelope/" ');
```

ニックネーム TEMPLATE オプションは、列 SYMBOL に [1,1] 指定が含まれるため、これを必須入力列として指定します。ニックネーム TEMPLATE オプションでは、全部の SOAP エンベロープが Web サービスに指定されています。getQuote 入力値は、SOAP エンベロープおよびボディ・エレメントに含まれています。

XPATH ニックネーム・オプションには、SOAP エンベロープおよびボディ・タグを通して取引価格を検索するための情報が含まれています。

| 以下の照会のように、"stockquote.stockquoteport_getquote_nn" ニックネームを使用して Web サービスにアクセスします。

```
| SELECT * FROM "stockquote.stockquoteport_getquote_nn"  
| WHERE symbol='IBM';
```

| シンボルが必須入力列であるため、このステートメントでは述部 symbol='IBM' を使用する必要があります。入力列では、等価述部のみが有効な述部です。等価述部のそれぞれが、入力メッセージ中の値を設定します。入力列がオプションである場合には、その列上に等価述部は必要ではありません。入力列が必須である場合、等価述部で照会を発行する必要があります。等価式に IBM などのリテラル値、あるいは、中間結合表またはニックネームからの値を使用することができます。

| 例 2: 繰り返しエレメントおよび子ニックネーム

| 以下の例では、getZooReport という名前の動物園のレポートを作成する Web サービスを使用します。入力値は、動物園 ID です。出力値は、以下のスキーマで表されるレポートです。

```

<wsdl:definitions name="Name"
  targetNamespace="http://myzoo.com"
  ...
<wsdl:types>
  <xsd:schema elementFormDefault="qualified" targetNamespace="http://myzoo.com"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="Animal">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:Species"/>
          <xsd:element ref="tns:Lot"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="AnimalCareList">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:Animal"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Count" type="xsd:string"/>
    <xsd:element name="LastModified" type="xsd:string"/>
    <xsd:element name="Lot" type="xsd:string"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="NumberFingersLeft" type="xsd:string"/>
    <xsd:element name="Species" type="xsd:string"/>
    <xsd:element name="Zoo">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:ZooName"/>
          <xsd:element ref="tns:Count"/>
          <xsd:element ref="tns:LastModified"/>
          <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Zookeeper"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="ZooName" type="xsd:string"/>
    <xsd:element name="Zookeeper">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="tns:Name"/>
          <xsd:element ref="tns:NumberFingersLeft"/>
          <xsd:element ref="tns:AnimalCareList"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="optional"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
  ...

```

図 34. *getZooReport* Web サービス

以下に示すのは、スキーマを含む WSDL に基づいて DB2 コントロール・センターの Discover ツールにより生成された DDL です。

```

CREATE NICKNAME zooport_getzooreport_nn (
  zooid VARCHAR (48) OPTIONS(TEMPLATE '&column'),
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@ns1:id'),
  report_zooname VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),
  report_count VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/ns1:Count/text()'),
  report_lastmodified VARCHAR (48)
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),
  zooport_getzooreport_pkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort' ,
  SOAPACTION 'http://myzoo.com/getZooReport' ,
  TEMPLATE '<soapenv:Envelope>
    <soapenv:Body>
      <zooId>&zooId[1,1]</zooId>
    </soapenv:Body>
  </soapenv:Envelope>' ,
  XPATH '/soapenv:Envelope/soapenv:Body' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com " ');

```

図 35. 動物園レポート - 親ニックネーム - zooport_getzooreport_nn

```

CREATE NICKNAME zooport_getzooreport_report_nn (
  zooport_getzooreport_fkey VARCHAR (16)
    FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
  zookeeper_id VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/@ns1:id'),
  report_name VARCHAR (48) OPTIONS(XPATH './ns1:Zookeeper/ns1:Name/text()'),
  report_numberfingersleft VARCHAR (48)
    OPTIONS(XPATH './ns1:Zookeeper/ns1:NumberFingersLeft/text()'),
  zooport_getzooreport_pkey VARCHAR (16)
    FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zoo' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

図 36. 動物園レポート - ニックネーム zooport_getzooreport_nn の子

```

CREATE NICKNAME zooport_getzooreport_report_report_nn (
  zooport_getzooreport_fkey VARCHAR (16) FOR BIT DATA NOT NULL
    OPTIONS(FOREIGN_KEY 'zooport_getzooreport_report_nn'),
  report_name VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Name/text()'),
  report_species VARCHAR (48)
    OPTIONS(XPATH './ns1:Animal/ns1:Species/text()'),
  report_lot VARCHAR (48) OPTIONS(XPATH './ns1:Animal/ns1:Lot/text()'))
FOR SERVER "zooserver" OPTIONS(
  XPATH './ns1:Zookeeper/ns1:AnimalCareList' ,
  NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/" ,
              ns1="http://myzoo.com" ');

```

図 37. 動物園レポート - zooport_getzooreport_report_nn の子

スキーマには、繰り返しエレメントまたはシーケンス・エレメントが含まれます。449 ページの図 35、449 ページの図 36、および 449 ページの図 37で示されているように、これらの繰り返しエレメントが親ニックネームの子ニックネームになります。たとえば、zooname、count、lastmodified、および zookeeper はすべて zoo のエレメントです。エレメント zoo には、0 またはそれ以上の zookeeper エレメントが含まれます。ルート・ニックネーム zoo には列、zooname、count、および lastmodified が含まれます。子ニックネーム zookeeper は、DB2 コントロール・センターの Discover ツールによって作成され、zookeeper の繰り返しエレメントを説明します。zookeeper 列の 3 番目のエレメント animalcarelist も 0 またはそれ以上のエレメントを含むため、子ニックネーム zooport_getzooreport_report_report_nn になります。次の図では、ニックネーム階層を示します。

ルート・ニックネーム: zooport_getzooreport_nn

Zoo (親):

- ZooName
- Count
- LastModified
- ZooKeeper (0 またはそれ以上の ZooKeeper エレメントがある)

子ニックネーム: zooport_getzooreport_report_nn

ZooKeeper エレメント

- Name
- NumberFingersLeft
- AnimalCareList (0 またはそれ以上の Animal エレメントがある)

子ニックネーム: zooport_getzooreport_report_report_nn

Animal

- Name
 - Species
 - Lot
-

図 38. 親 → 子 → ニックネーム階層

以下のステートメントは、動物園レポートの Web サービスにアクセスするためニックネームに発行する照会の標準的なものです。このステートメントを発行する際には、特定の ID に基づき、また子ニックネームの動物園レポートの主キーと外部キーがどこで一致するかに基づいて動物園レポートからの情報を検索します。

```
SELECT * FROM zooport_getzooreport_nn ,
         zooport_getzooreport_report_nn zk ,
         zooport_getzooreport_report_report_nn a
WHERE zooid='1'AND zooport_getzooreport_pkey=zk.zooport_getzooreport_fkey
and zk.zooport_getzooreport_pkey=a.zooport_getzooreport_fkey;
```

例 3: 遅延バインディング

以下の例では、Late Binding オプションを使用する方法を示します。このオプションは、DB2 コントロール・センターまたは DB2 コマンド行から使用できます。ニックネーム・オプション URL および SOAPACTION を定義し、ニックネーム作成時に列オプション URLCOLUMN および SOAPACTIONCOLUMN を使用可能にし

た場合、遅延バインディング機能を使用していることとなります。DB2 コントロール・センターは 2 つの列オプション URLCOLUMN および SOAPACTIONCOLUMN を作成し、それらの列の値を yes に設定します。

以下の例は、ある会社のためにすべてのサプライヤーがインプリメントする部品の見積価格を提供する Web サービスのものです。ここに、URLCOLUMN および SOAPACTIONCOLUMN 定義を含む CREATE ステートメントがあります。

```
CREATE NICKNAME GetPartQuote(  
  partnumber INTEGER OPTIONS (TEMPLATE'&column'),  
  price FLOAT OPTIONS (XPATH './price'),  
  urlcol VARCHAR(100) OPTIONS (URLCOLUMN 'Y'),  
  soapactioncol VARCHAR(100) OPTIONS (SOAPACTIONCOLUMN 'Y'),  
  FOR SERVER myServer  
  OPTIONS (  
    ...  
    SOAPACTION 'http://example.com/GetPartPrice' ,  
    URL 'http://mycompany.com:9080/GetPartPrice',  
    ...  
  )
```

単一の照会ですべてのサプライヤーからの見積価格を得るには、SOAPACTION および URL 列オプションにそれぞれのサプライヤーが使用する値が必要です。照会は次のようになります。

```
SELECT * FROM supplier_endpoints p,  
  GetPartQuote q  
  WHERE partnumber=1234 AND  
    p.url=q.urlcol AND  
    p.soapaction=q.soapactioncol;
```

ローカル表 supplier_endpoints には、Web サービスを呼び出すのに使用するすべての URL および SOAP が含まれています。ORDER BY 価格文節を含めて、この部品が最も安価なサプライヤーを判別することができます。

例 4: ESCAPE_INPUT 列オプション

照会に、入力値として XML フラグメントを含めることができます。ニックネームを登録する際に、列オプション ESCAPE_INPUT=N を含めてください。このオプションは < や > などの特殊文字を入力値の XML フラグメント中に維持します。

SOAP メッセージの一部として XML を送信することが必要である繰り返し入力値がスキーマに含まれている場合、ESCAPE_INPUT 列オプションを使用して、正しい XML で出力メッセージを作成することができます。

たとえば、動物園 Web サービスには新しい動物園の飼育係とその飼育係と関連した動物を追加する操作が含まれています。この例のスキーマでは、AnimalCareList は複数の動物を持つことができます。

```
CREATE NICKNAME add_zookeeper(  
  zookeeper_id VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  name VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  numberfingersleft VARCHAR(48) OPTIONS(TEMPLATE '...'),  
  animals VARCHAR(3000) OPTIONS( TEMPLATE '...' , ESCAPE_INPUT 'N')  
  ...
```

新しい動物園の飼育係を 2 匹の動物と共に追加するには、以下のような照会を発行します。

```

SELECT * FROM add_zookeeper
  WHERE zookeeper_ID='37' AND
         name='Amit Tsunami' AND
         numberfingersleft='3' AND
         animals='<AnimalCareList xmlns="http://myzoo.com">
                   <Animal>
                     <Name>Larry</Name>
                     <Species>Gorilla</Species>
                     <Lot>7</Lot>
                   </Animal>
                   <Animal>
                     <Name>Bill</Name>
                     <Species>Chimpanzee</Species>
                     <Lot>8H</Lot>
                   </Animal>
                 </AnimalCareList>';

```

add_zookeeper ニックネームは、Web サービスの状態を変更または情報を更新することができる Web サービス操作です。非リレーショナル・ラッパーを更新することはできませんが、この例の SELECT ステートメントは新しい動物園の飼育係を追加して動物園の情報を更新します。

xsd:anyType などのエレメントを使用するスキーマには、ESCAPE_INPUT 列オプションを使用することもできます。この場合、エレメントのタイプは知られていません。入力に対して任意の XML フラグメントを指定できるよう、そのエレメントには ESCAPE_INPUT 列オプションを入力列に使用できます。

関連概念:

- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 460 ページの『Web サービス・ラッパーのメッセージ』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項

等価述部

入力列で有効な述部は等価述部のみです。出力列では、どの述部も有効です。

以下の例では、この列では述部がサポートされていないことを示すメッセージと共にエラーが戻されます。この例では、列 zipcode は入力列です。

```
SELECT return FROM gettemp WHERE zipcode<'95141'
```

以下の例では、入力列に等価述部を使用した有効な照会を示します。カスタマーのニックネームは、カスタマー ID を含むローカル DB2 UDB 表で結合されています。照会には出力のみの列である Sales 列に追加の述部が含まれています。

```
SELECT a.name, a.address
       FROM customers a, local_table b
       WHERE
         a.customer_id=b.custid AND
         a.Sales > 300000;
```

必須入力列の述部

参照するニックネーム階層の SQL 照会では、すべての必須入力列に等価述部値を提供しなければなりません。ラッパーは、この制限に違反するすべての照会に対して、SQLCODE 901 を戻します。

IN または OR 述部

WebSphere Business Integration ラッパーおよび Web サービス・ラッパーでは、入力列に対して IN または OR 述部は許可されていません。

以下の例は、無効な照会を示しています。カスタマーのニックネームには 1 つの必須列 customer_id が含まれています。

```
SELECT * FROM customers
       WHERE customer_id IN (12345, 67890, 11223);
SELECT * FROM customers
       WHERE customer_id IN (SELECT custid FROM local_table); )
```

ただし、WebSphere Business Integration ラッパーの場合、必須入力列に SPECIFICATION ONLY パラメーターでユニーク索引を定義した場合、IN リスト述部を必須入力列と共に使用できます。

```
CREATE UNIQUE INDEX myuindex ON customers(customer_id) SPECIFICATION ONLY;
```

オプション入力列の結合

以下の例では、オプション入力列の結合の制限を説明します。オプション入力列をローカル表またはニックネームから結合することはできません。WSDL がオプションとして入力ニックネーム列を生成し、その列を結合に使用する必要がある場合には、DDL を編集してその列を必須入力列に変更しなければなりません。

この例では、order という名前の Web サービス・ラッパー・ニックネームが、shipping_method をオプション入力列として作成されます。以下のステートメントは、述部でリテラルを使用するため、有効な照会です。

```
SELECT * FROM order
       WHERE part="hammer" AND shipping_method="FEDEX";
```

ただし、部品と配送方法を定義する orderparts という名前のローカル表を照会に含め、その表にオプションの shipping_method という列が含まれる場合、ステートメントは無効です。

```
SELECT * FROM
       order o, orderparts op
       WHERE
         o.part="hammer" AND
         o.shipping_method=op.shipping_method
```

WebSphere Business Integration ラッパーの場合、ニックネームのオプション入力列の述部がWebSphere Business Integration アダプターにプッシュ・ダウンされる可能性があります。DB2 UDB は、アプリケーション・データ・ソースから取り出された行にそれらの述部をローカルに適用することができます。所定の入力列の述部が常にアダプターへプッシュ・ダウンされるようにするには、入力列を必須入力列と宣言してください。それぞれのニックネーム階層への照会には、必須入力列の述部値が含まれていなければなりません。

有効な結果を得るためには、結合された入力列は、Web サービス・ラッパーの必須列である必要があります。

外部結合

親ニックネームからの主キーおよび子ニックネーム列からの外部キーを使用したニックネーム間の外部結合はサポートされていません。

XML 文書中の親エレメントに子エレメントが含まれておらず、親ニックネームと子ニックネーム間で内部結合を使用した場合、そのエレメントには行が戻されません。たとえば、指定の顧客の bankdetail 情報が SAP システムにない場合、その特定の顧客の場合、sap_bapi_customer_getdetail2_sap_customerbankdetail_NN nickname に行は戻されません。

以下の CREATE NICKNAME ステートメントは、例の照会で使用される列を定義します。

```
CREATE NICKNAME sap_bapi_customer_getdetail2_NN(
  ...
  NAME VARCHAR(35)
    OPTIONS(XPATH './ns3:sap_customeraddress/
      ns1:sap_customeraddress/ns1:NAME/text()'),
  ...
  NN_PKEY VARCHAR(16) OPTIONS(PRIMARY_KEY 'YES'),
  COMPANYCODE VARCHAR(4) OPTIONS(XPATH './ns3:COMPANYCODE/text()'),
    TEMPLATE '<ns3:COMPANYCODE>&column</ns3:COMPANYCODE>'),
  CUSTOMERNO VARCHAR(10) OPTIONS(XPATH './ns3:CUSTOMERNO/text()'),
    TEMPLATE '<ns3:CUSTOMERNO>&column</ns3:CUSTOMERNO>'),
  ...
  FOR SERVER sap_server
  OPTIONS(XPATH '//ns3:sap_bapi_customer_getdetail2',
    TEMPLATE '<ns3:sap_bapi_customer_getdetail2>
      &sap_bapi_customer_getdetail2_sap_customerbankdetail_NN[0,1]
      &COMPANYCODE[0,1]
      &CUSTOMERNO[1,1]
    </ns3:sap_bapi_customer_getdetail2>',
  ...
)
```

図 39. getdetail2 ニックネームからの抜粋

```

CREATE NICKNAME sap_bapi_customer_getdetail2_sap_customerbankdetail_NN(
  CUSTOMER VARCHAR(10)  OPTIONS(XPATH './ns2:CUSTOMER/text()'),
    TEMPLATE '<ns2:CUSTOMER>&column</ns2:CUSTOMER>'),
  BANK_KEY VARCHAR(15)  OPTIONS(XPATH './ns2:BANK_KEY/text()'),
    TEMPLATE '<ns2:BANK_KEY>&column</ns2:BANK_KEY>'),
  BANK_ACCT VARCHAR(18)  OPTIONS(XPATH './ns2:BANK_ACCT/text()'),
    TEMPLATE '<ns2:BANK_ACCT>&column</ns2:BANK_ACCT>'),
  CTRL_KEY VARCHAR(2)   OPTIONS(XPATH './ns2:CTRL_KEY/text()'),
    TEMPLATE '<ns2:CTRL_KEY>&column</ns2:CTRL_KEY>'),
  BANK_REF VARCHAR(20)  OPTIONS(XPATH './ns2:BANK_REF/text()'),
    TEMPLATE '<ns2:BANK_REF>&column</ns2:BANK_REF>'),
  NN_FKEY VARCHAR(16)  OPTIONS(FOREIGN_KEY 'SAP_BAPI_CUSTOMER_GETDETAIL2_NN'))
FOR SERVER sap_server
OPTIONS(XPATH './ns3:sap_customerbankdetail/ns2:sap_customerbankdetail',
  TEMPLATE '<ns3:sap_customerbankdetail>
  <ns2:sap_customerbankdetail>
    &CUSTOMER[0,1]
    &BANK_KEY[0,1]
    &BANK_ACCT[0,1]
    &CTRL_KEY[0,1]
    &BANK_REF[0,1]
  </ns2:sap_customerbankdetail>
</ns3:sap_customerbankdetail>',
  ...

```

図 40. カスタマー銀行詳細ニックネームからの抜粋

以下の例では、2 つのニックネーム間に内部結合条件があるため、照会は行を戻しません。

```

SELECT a.name, b.bank_key
FROM sap_bapi_customer_getdetail2_NN a,
     sap_bapi_customer_getdetail2_sap_customerbankdetail_NN b
WHERE a.customerno='1234567890'
AND a.NN_PKEY=b.NN_FKEY;

```

WebSphere Business Integration ラッパーまたは Web サービス・ラッパーのニックネーム定義に必須入力列が含まれる場合、このニックネームと他のローカル DB2 UDB 表または他のニックネームとの間の左外部結合はサポートされません。

関連概念:

- 149 ページの『ニックネームおよび列レベルでの TEMPLATE オプション』

関連タスク:

- 142 ページの『ビジネス・アプリケーション・データ・ソースのフェデレーテッド・システムへの追加』
- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』

関連資料:

- 173 ページの『ビジネス・アプリケーション・データ・ソース - 照会の例』
- 156 ページの『CREATE NICKNAME ステートメント - WebSphere Business Integration ラッパーの例』
- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』

Web サービス・データ・ソース - 照会の例

例 1: マテリアライズ照会表の使用

マテリアライズ照会表は、照会の結果をローカルにキャッシュに入れて、照会のパフォーマンスを向上させるために使用します。 Web サービス・データ・ソースからのニックネームを使用して、マテリアライズ照会表を作成することができます。照会によっては、マテリアライズ照会表が基本表にアクセスしないで照会に応答できるかを、データベースは自動的に判別できます。以下のプロシーチャーは、マテリアライズ照会表の作成およびデータ設定の方法を示します。

1. ローカルまたは基本表を、次のようにして作成します。

```
CREATE TABLE mystocks(ticker VARCHAR(10));
```

キャッシュに入れるすべての値の維持に、ローカル表を使用できます。

2. キャッシュに入れるすべての値を表に挿入します。

```
INSERT INTO mystocks VALUES('IBM');
INSERT INTO mystocks VALUES('MSFT');
...
```

3. Web サービス・ニックネームを次のようにして作成します。

```
CREATE NICKNAME stockquote_nn (
    ticker VARCHAR(40) OPTIONS (TEMPLATE '&column'),
    price VARCHAR(16) OPTIONS (XPATH './Result/text()')
)
FOR SERVER stock_server
OPTIONS (TEMPLATE '<ticker>&column</ticker>'
        XPATH './Result/text()');
```

4. ニックネームおよびローカル表から成るビューを、次のようにして作成します。

```
CREATE VIEW stock_quote_view (ticker, price)
AS (
    SELECT nn.ticker, nn.price
    FROM stockquote_nn nn, mystocks s
    WHERE nn.ticker=s.ticker
);
```

5. マテリアライズ照会表を次のようにして作成します。

```
CREATE TABLE stockquote_MQT (ticker, ticker2, price)
as (SELECT nn.ticker,s.ticker as ticker2, nn.price
FROM stockquote_nn nn, mystocks s
WHERE nn.ticker=s.ticker )
DATA INITIALLY DEFERRED REFRESH DEFERRED;
```

マテリアライズ照会表の出カリストに結合述部 (nn.ticker and s.ticker) で使用されている VARCHAR 列のすべてを含め、マテリアライズ照会表が DB2 Universal Database に使用される機会を最大化します。

マテリアライズ照会表の最新表示を据え置くには、REFRESH DEFERRED キーワードを指定します。REFRESH DEFERRED キーワードで指定されたマテリアライズ照会表は、基礎となる基本表への変更を反映しません。文節 DATA INITIALLY DEFERRED を使用して、データがCREATE TABLE ステートメントの一部として表に挿入されないようにします。

6. REFRESH TABLE ステートメントを発行して表にデータを追加します。表中のデータは、REFRESH TABLE ステートメントの発行時の照会の結果をスナップショットとして反映します。以下の例では、stockquote_MQT 表にデータを追加し、現在の最新表示経過時間の特殊レジスターの値を設定します。

```
REFRESH TABLE stockquote_MQT;  
  
SET CURRENT REFRESH AGE any;
```

実行先がマテリアライズ照会表のデータである照会は、実行先が基本表のデータである照会よりも高速です。マテリアライズ照会表を使用したい場合、以下のようにニックネームではなくビューを参照します。

```
SELECT * FROM stock_quote_view  
WHERE ticker='IBM';
```

キャッシュに入れられていない値を選択する照会を発行した場合、0 行が戻されません。

例 2: 主キーと外部キーを使用した結合の実行

PRIMARY_KEY および FOREIGN_KEY 列は、親ニックネームと子ニックネームの間の関係を定義するために使用されます。各親ニックネームは、主キー列オプションをもっていなければなりません。ある親ニックネームの子は、親ニックネームの主キー列を参照する外部キー列オプションによって定義します。1 つのニックネームに対して複数の子は可能ですが、親については 1 つのみ可能です。

これらの列にはバイナリー・データのみが含まれるため、列は FOR BIT DATA NOT NULL キーワードで定義されています。DB2 コントロール・センターは、ニックネームを作成する時にこの定義を生成します。ニックネームを作成する時に、PRIMARY_KEY および FOREIGN_KEY 列を FOR BIT DATA NOT NULL として明示的に定義することもできます。

以下の例では、Web サービス・ラッパーが PRIMARY_KEY および FOREIGN_KEY 列を使用して親ニックネームと子ニックネームを関連付ける方法を示します。

```
CREATE NICKNAME zooport_getzooreport_nn (  
  zooid VARCHAR (48) OPTIONS(TEMPLATE '&column'),  
  zoo_id VARCHAR (48) OPTIONS(XPATH './ns1:Zoo/@id'),  
  report_zoo_zooname VARCHAR (48)  
    OPTIONS(XPATH './ns1:Zoo/ns1:ZooName/text()'),  
  report_zoo_count VARCHAR (48)  
    OPTIONS(XPATH './ns1:Zoo/ns1:Count/text()'),  
  report_zoo_lastmodified VARCHAR (48)  
    OPTIONS(XPATH './ns1:Zoo/ns1:LastModified/text()'),  
  nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES'),  
  url VARCHAR (256) OPTIONS(URLCOLUMN 'Y'),  
  soapaction VARCHAR (256) OPTIONS(SOAPACTIONCOLUMN 'Y')  
) FOR SERVER "mytestsvr"  
OPTIONS(  
  URL 'http://localhost:9080/MaelstromTest/services/ZooPort',  
  SOAPACTION 'http://myzoo.com/getZooReport',  
  TEMPLATE '<soapenv:Envelope>  
    <soapenv:Body>  
      <zooId>&zooId[1,1]</zooId>  
    </soapenv:Body>  
  </soapenv:Envelope>'
```

```

        XPATH '/soapenv:Envelope/soapenv:Body' ,
        NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
                    ns1="http://myzoo.com" ');

CREATE NICKNAME zooport_getzooreport_report_zookeeper_nn (
    nn_fk VARCHAR (16) NOT NULL
    OPTIONS(FOREIGN_KEY 'ZOOPORT_GETZOOREPORT_NN'),
    zookeeper_id VARCHAR (48) OPTIONS(XPATH './@id'),
    report_zookeeper_name VARCHAR (48) OPTIONS(XPATH './ns1:Name/text()'),
    zookeeper_numberfingersleft VARCHAR(48)
    OPTIONS(XPATH './ns1:NumberFingersLeft/text()'),
    nn_pk VARCHAR (16) NOT NULL OPTIONS(PRIMARY_KEY 'YES')
)
FOR SERVER "MYTESTSRVR" OPTIONS(
    XPATH './ns1:Zoo/ns1:Zookeeper' ,
    NAMESPACES ' soapenv="http://schemas.xmlsoap.org/soap/envelope/",
                ns1="http://myzoo.com" ');

```

ニックネーム `zooport_getzooreport_report_zookeeper_nn` の外部キー `nn_fk` は、外部キー・オプションの親ニックネーム `zooport_getzooreport_nn` を指します。指定された主および外部キー・ニックネーム列は、WSDL 文書中のデータには対応しません。というのは、これらのニックネーム列には、ラッパーによって生成されたキーが含まれるからです。これらのキーは、照会内でのみ固有である親と子ニックネーム間の関係を識別します。子ニックネームが入力列を含む場合、親ニックネーム・オプション・テンプレートは、ニックネーム・テンプレート構造内で、その子ニックネームを参照します。

以下の SQL ステートメントは親ニックネームと子ニックネームを結合します。

```

SELECT *
FROM   zooport_getzooreport_nn a,
       zooport_getzooreport_report_zookeeper_nn z,

WHERE  a.nn_pk = z.nn_fk
       AND a.zooid = 100
;

```

以下の記述では、照会実行時に Web サービス・ラッパーが `TEMPLATE` および `XPATH` ニックネームと列オプションを使用する方法を説明します。これは特定のインプリメンテーションの例として意図されたものではありません。

主キーと外部キー列を結合すると、Web サービス・ラッパーが Web サービス・プロバイダーへメッセージを送信し、Web サービス・プロバイダーからは行のセットが戻されます。ラッパーは、列オプション・テンプレート (`ZOOID VARCHAR (48) OPTIONS(TEMPLATE '&column')`) への参照の照会からの入力列の値 (`a.zooid = 100`)、それからニックネーム・テンプレート・オプション (`<zooId>&zooId[1,1]</zooId>`) 中のすべての列参照を置き換えることにより、親ニックネームのメッセージを生成します。ニックネーム・テンプレート・オプションには列参照または子ニックネーム参照が含まれることがあります。その後メッセージは Web サービスに送信されます。

ラッパーは、Web サービスが戻す文書にニックネーム・オプション `XPATH` を適用することにより、ニックネームの行を生成します。ニックネーム・オプション `XPATH` が複数の XML フラグメントを戻す場合には、ニックネームには複数の行が含まれます。列 `XPATH` オプションが、行を表す結果の XML フラグメントに適用され、列値を得ます。ニックネームに 1 つ以上の間接的な親がある場合、親ニッ

クネーム XPATH 式のすべてが、階層のトップから、このニックネームにニックネーム・オプション XPATH および列オプション XPATH が適用される手前まで、順番に適用されます。

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データのソースの追加』
- 434 ページの『Web サービス・ラッパーの登録』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 460 ページの『Web サービス・ラッパーのメッセージ』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

Web サービス・ラッパーのセキュリティ

Web サービス・ラッパーは、SOAP メッセージのトランスポート・プロトコルとして HTTPS をサポートします。HTTPS は、多くの Web サービス・プロバイダーが使用している標準の暗号化プロトコルです。Web サービス・プロバイダーが生成する WSDL 文書には、URL に `https://` が含まれています。HTTP 要求または HTTP 応答中の SOAP メッセージは暗号化されます。

Web サービスがトランスポート・プロトコルとして HTTPS を使用する場合、Web サービス・ラッパーは、識別のためにサーバーが送信する SSL 証明書を検証しません。Web サービス・ラッパーは、自己署名証明書で Web サービスを呼び出すことができます。

Web サービス・ラッパーは、CREATE USER MAPPING ステートメントを使用して、HTTP 認証をサポートします。CREATE USER MAPPING ステートメントを使用して、フェデレーテッド・サーバーのユーザー ID を Web サービスのユーザー ID およびパスワードにマップしてください。ラッパーの開発者は、ユーザー ID およびパスワードに、以下の例のようなユーザー・マッピング・ステートメントを提供します。

```
CREATE USER MAPPING
  FOR RSPALTEN SERVER S1
  OPTIONS ( REMOTE_AUTHID 'SYSTEM', REMOTE_PASSWORD 'MANAGER' )
```

S1 サーバー上の Web サービス・ニックネームがアクセスされると、HTTP 要求が SYSTEM というユーザー ID、MANAGER というパスワードで送信されます。ユーザー・マッピングは任意です。ユーザー・マッピングを指定しない場合、Web サービス・プロバイダーが認証情報を期待する際にエラーが出る可能性があります。サーバーの中には、サービスへのアクセスを制約するために認証を使用するものがあります。認証が必要であるかは、WSDL 文書中の情報からは明らかではありません。

関連概念:

- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 434 ページの『Web サービス・ラッパーの登録』
- 436 ページの『Web サービス・データ・ソースのニックネームの登録』

関連資料:

- 460 ページの『Web サービス・ラッパーのメッセージ』

Web サービス・ラッパーのメッセージ

以下のテーブルでは、Web サービス・ラッパー使用時に受信する可能性のある標準的なエラー・メッセージのいくつかを説明します。

表 90. 標準的なエラー・メッセージ

エラー	説明	ユーザーの対応
SQL1822N データ・ソース "wswrap.svl.ibm" から予期しないエラー・コード "SOAP-Fault" を受け取りました。関連するテキストおよびトークンは "java.lang.Exception: HTTP URI s" です。 SQLSTATE=560BD	Web サービス・プロバイダーが SOAP 障害を戻しました。	<ul style="list-style-type: none"> • 入力に誤りである可能性があります。入力引き数が正しいことを確認してください。 • Web サービス・プロバイダーに問題がある可能性があります。Web サービスの所有者に連絡してください。
SQL30081N 通信エラーが検出されました。使用されている通信プロトコル: "SOAP"。使用されている通信 API: "HTTP"。エラーが検出された位置: "localhos"。エラーを検出した通信関数: "connect"。プロトコルに特定のエラー・コード: "38309"、"10061"、"1"。 SQLSTATE=08001	Web サービス・プロバイダーがポートまたは URL で聴取しませんでした。	<ul style="list-style-type: none"> • 使用している URL がすべて有効であるか確認します。 • ポート情報が正しいかどうかを確認します。 • サーバー稼働していることを確認します。
SQL30081N 通信エラーが検出されました。使用されている通信プロトコル: "SOAP"。使用されている通信 API: "HTTP"。エラーが検出された位置: "doesntexist.ibm"。エラーを検出した通信関数: "*"。プロトコルに特定のエラー・コード: "38308"、"*"、"0"。 SQLSTATE=08001	Web サービス・プロバイダー・ホスト名がドメイン・ネーム・サーバーにありません。	<ul style="list-style-type: none"> • ホスト名がネーム・サーバーにあることを確認してください。

表 90. 標準的なエラー・メッセージ (続き)

エラー	説明	ユーザーの対応
SQL30081N 通信エラーが検出されました。使用されている通信プロトコル: "SOAP"。使用されている通信 API: "HTTP"。エラーが検出された位置: "www.ibm.com"。エラーを検出した通信関数: "*"。プロトコルに特定のエラー・コード: "38312"、"*"、"0"。SQLSTATE=08001	ブラウザが通常報告する 404 エラーに類似した HTTP 戻りコード。	サーバーがアプリケーションにエラーを戻しているかを判別します。DB2 UDB トレースを実行して、サーバーがどのような応答を戻しているかを判断します。
SQL30081N 通信エラーが検出されました。使用されている通信プロトコル: "SOAP"。使用されている通信 API: "HTTP"。エラーが検出された位置: ""。エラーを検出した通信関数: "*"。プロトコルに特定のエラー・コード: "38304"、"*"、"0"。SQLSTATE=08001	URL が誤りです。SQLSTATE 38304 はプロトコルが既知でないことを示している可能性があります。SQLSTATE 38305 は URL 中の構文エラーを示します。	WSDL 文書に有効な URL 構文およびプロトコルが含まれることを確認します。
SQL1822N データ・ソース "wswrap.svl.ibm" から予期しないエラー・コード "SAXException" を受け取りました。関連するテキストおよびトークンは "Expected end of tag 'ns0:Mi'" です。SQLSTATE=560BD	応答には XML 出力上の構文解析エラーが含まれます。	サーバーが正しい XML 出力を戻したことを確認してください。DB2 UDB トレースを実行して、Web サービス・プロバイダーが戻す応答を判断します。異なるツールで Web サービスを呼び出し、Web サービスの応答が妥当であることを確認することもできます。

関連概念:

- 427 ページの『Web サービス・ラッパーおよび Web サービス記述言語文書』

関連タスク:

- 433 ページの『フェデレーテッド・サーバーへの Web サービス・データ・ソースの追加』
- 434 ページの『Web サービス・ラッパーの登録』

関連資料:

- 446 ページの『CREATE NICKNAME ステートメント - Web サービス・ラッパーの例』
- 456 ページの『Web サービス・データ・ソース - 照会の例』
- 169 ページの『ビジネス・アプリケーション・ラッパーおよび Web サービス・ラッパーの照会制約事項』

第 24 章 XML データ・ソースへのアクセスの構成

この章では、XML データ・ソースに保管されているデータにアクセスできるようにフェデレーテッド・サーバーを構成する方法について説明します。XML データ・ソースへのアクセスは、DB2 コントロール・センターを使用するか、または SQL ステートメントを発行して構成できます。

この章では、

- XML の概要を説明します。
- 実行する必要があるタスクがリストされています。
- 必要な SQL ステートメントの例を挙げます。
- XML ラッパーに関連するエラー・メッセージをリストします。

XML とは?

Extensible Markup Language (XML) は、構造化された文書およびデータ用の汎用フォーマットです。XML ファイルは `xml` のファイル拡張子をもっています。HTML のように、XML は文書の中のデータを構造化するために、タグ (`<` および `>` の括弧に入れられた語) を使用します。サンプル XML 文書が 図 41 に示されています。

```
<?xml version="1.0" encoding=UTF-8"?>
<doc>
  <customer id='123'>
    <name>...</name>
    <address>...</address>
    ...
    <order>
      <amount>...</amount>
      <date>...</date>
      <item quant='12'>
        <name>...</name>
      </item>
      <item quant='4'>...</item>
      ...
    </order>
    <order>...</order>
    ...
    <payment>
      <number>...</number>
      <date>...</date>
    </payment>
    <payment>...</payment>
    ...
  </customer>
  <customer id='124'>...</customer>
</doc>
```

図 41. サンプル XML 文書

XML ラッパーがどのように機能するか

XML ラッパーにより、次のような種類のデータを照会するために SQL を使用できるようになります。

- 1 個のファイルとして保管されている複数の外部 XML 文書
- 1 つのディレクトリー・パス内にある複数の XML ファイル
- URI (Uniform Reference Identifier) で参照されるリモート XML ファイル
- リレーショナル列に保管された XML 文書

図 42 に、XML ラッパーによるフェデレーテッド・システムの処理方法を示します。

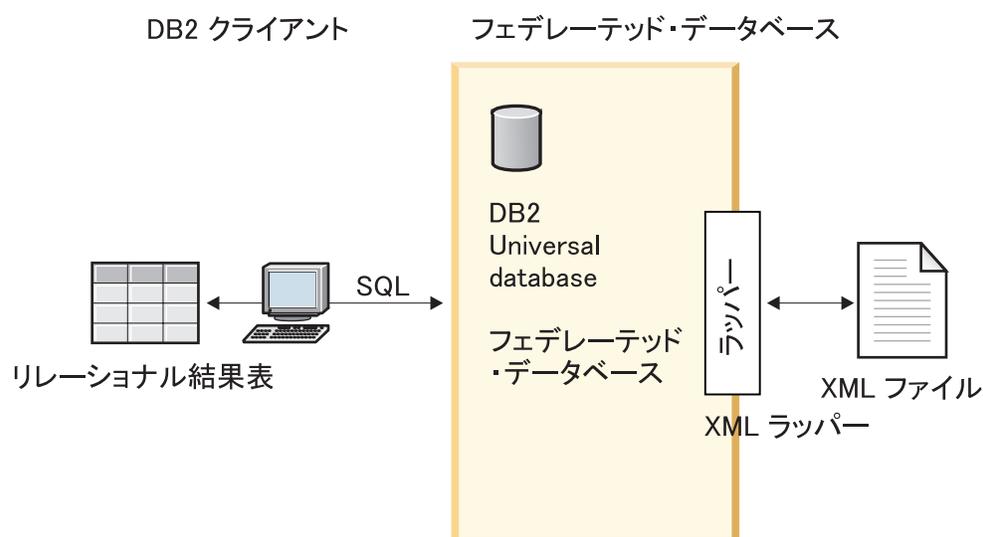


図 42. XML ラッパーがどのように機能するか

XML ラッパーにより、外部データ・ソースからの XML データから、ニックネームのセットで構成されるリレーショナル・スキーマへのマッピングが可能になります。XML 文書の構造は、ネストされていて繰り返すエレメントが、外部キーをもっている別の表としてモデルとなっているリレーショナル・スキーマと論理的には同等です。

XML 文書に対応するニックネームは、親ニックネームと子ニックネームの関係が、エレメントのネスト関係としてマッピングされるツリー構造に編成されています。

ネスト・エレメントが繰り返されている場合や、またはそれぞれ性質の異なるものが複雑な構造になっている場合には、それらネスト・エレメントのそれぞれに別個のニックネームを付けることができます。

子および親ニックネームは、ラッパーによって生成された主および外部キーによって接続されます。

XPath 式は、XML 文書をニックネームのセットで構成されているリレーショナル・スキーマにマップするために使用されます。XPath は、XML ファイルの一部

(XML 文書ツリー内のノードおよび属性のグループなど) を識別するためのアドレッシング・メカニズムです。基本 XPath 構文はファイル・システム・アドレッシングに似ています。

各ニックネームは、個々のタプルを表す XML エlementを識別する 1 つの XPath 式、および各Elementから列値を取り出す方法を指定する XPath 式のセットによって定義されます。

XML 文書マッピングの例:

次の例は 463 ページの図 41 に示されているサンプル XML 文書が、ニックネームのセットにマップされる方法、主および外部キーを使用することによって親子関係が確立される方法、文書の各Element内の個々のタプルおよび列を定義するために XPath 式を使用する方法、ならびに XML 文書がいったんフェデレーテッド・システムに登録された後、その XML 文書に対して照会を実行する方法を示しています。

サンプル XML 文書には、customer Elementのセットが含まれています。その各Elementには、複数の order Elementおよび payment Elementが入れられます。

order Elementは、複数の item Elementを囲んでいます。

Element間の関係は 図 43 に示されています。

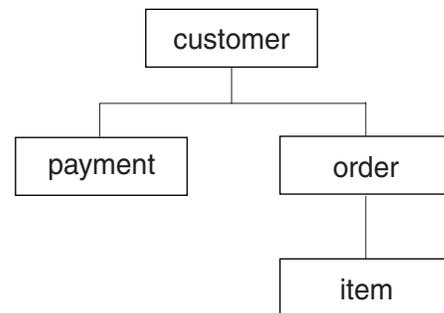


図 43. サンプル XML 文書のツリー構造

この構造に基づいて、CREATE NICKNAME ステートメントを使うことにより、XML 文書を、次の 4 つのニックネームが含まれるリレーショナル・スキーマにマッピングできます。

- customers
- orders
- payments
- items

ニックネーム間の関係は、特別な主および外部キー・ニックネーム列オプションを使用して、各ニックネームを親または子ニックネームとして指定することによって定義されます。各親ニックネームは、主キー列オプションによって指定された特別な列をもっていなければなりません。ある親ニックネームの子は、親ニックネームの主キー列を参照する外部キー列オプションによって定義します。指定された主お

よび外部ニックネーム列は、XML 文書の中のデータには対応しません。というのは、これらのニックネーム列には、ラッパーによって生成されたキーが含まれるからです。1 つのニックネームに対して複数の子は可能ですが、親については 1 つのみ可能です。ただし、ルート・ニックネームに親はありません。

サンプル XML 文書の場合、customers ニックネームに主キーが定義されており、orders、payments、および items ニックネームには、それぞれその親ニックネームを指す外部キーが定義されています。orders と payments のニックネームの外部キーは customers ニックネームを指し、items ニックネームの外部キーは orders ニックネームを指しています。

個々のタプルを表す XML エlementを識別するには、1 つの XPath 式を作成します。この例の場合、すべての customer エlementを参照するには '/doc/customer' XPath 式を使用し、すべての order エlementを参照するには './order' XPath 式を使用します。XPath 式 './order' に含まれるピリオドは、各 order エlementのタプルが、それに対応する customer エlementの各タプル内にネストされていることを示しています。

各Elementから列値を取り出す方法を指定するために、XPath 式のセットを作成します。この例の場合、customer エlementの id 属性 (この時点では、ニックネームの中に定義されている列) を参照するには './@id' XPath 式を使用します。customer エlementの name エlementを参照するには './name' XPath 式を使用し、customer エlementの address エlementを参照するには './address/@street' XPath 式を使用します。

CREATE NICKNAME ステートメントを使用して XML 文書をニックネームのセットにマップしたなら、文書の各Element内の個々のタプルおよび列を定義する XPath 式を使用し、主および外部キーを使用することによって、各ニックネームを親または子として定義します。その後、その XML 文書に対して SQL 照会を実行できます。

関連概念:

- 471 ページの『ニックネームと XML 文書間のデータ関連』

関連タスク:

- 466 ページの『フェデレーテッド・システムへの XML の追加』

フェデレーテッド・システムへの XML の追加

フェデレーテッド・システムへの XML の追加

フェデレーテッド・サーバーが XML データ・ソースにアクセスするように構成するには、アクセスするデータ・ソースとオブジェクトの情報をフェデレーテッド・サーバーに渡す必要があります。

| DB2 コントロール・センター、または DB2 コマンド行プロセッサを使用して、
| フェデレーテッド・サーバーが XML データ・ソースにアクセスするように構成で
| きます。DB2 コントロール・センターには、フェデレーテッド・サーバーの構成
| に必要な手順を示すウィザードが含まれています。

前提条件:

- DB2 Information Integrator が、フェデレーテッド・サーバーとして機能するサーバー上にインストールされている必要があります。
- フェデレーテッド・データベースはフェデレーテッド・サーバー上になければなりません。

手順:

XML データ・ソースをフェデレーテッド・サーバーに追加するには、以下のようになります。

1. XML ラッパーを登録します。
2. XML サーバー定義を登録します。
3. XML データ・ソースのニックネームを登録します。
4. 非ルート・ニックネームのフェデレーテッド・ビューを作成します。

| ルート・ニックネームは、ニックネーム階層の最上位にあるニックネームです。
| 非ルート・ニックネームは、ニックネーム階層内の親を持つニックネームです。
| XML 文書内で最上位の要素ではないルート・ニックネームも存在することがあります。
|

このステートメントは、DB2 コントロール・センターまたは DB2 コマンド行プロセッサから実行できます。XML ラッパーがフェデレーテッド・システムに追加された後、XML データ・ソースに対して照会を実行できます。

関連概念:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のインストール - 概要』

関連タスク:

- 57 ページの『フェデレーテッド・データベースの作成』
- 467 ページの『XML ラッパーの登録』
- 469 ページの『XML データ・ソースのサーバーの登録』
- 474 ページの『XML データ・ソースのニックネームの登録』
- 479 ページの『非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)』

関連資料:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (32 ビット)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator でサポートされるオペレーティング・システム (64 ビット)』

XML ラッパーの登録

XML ラッパーの登録は、XML データ・ソースをフェデレーテッド・サーバーに追加するという、より大きな作業の一部を成しています。

HMMER データ・ソースにアクセスするには、ラッパーを登録する必要があります。ラッパーは、データ・ソースとの通信やデータの検索のためにフェデレーテッド・サーバーによって使用されます。ラッパーは、ライブラリー・ファイルのセットとしてインプリメントされます。

手順:

ラッパーを登録するには、ラッパーの名前とラッパー・ライブラリー・ファイルの名前を指定して、CREATE WRAPPER ステートメントを発行します。

例えば、AIX オペレーティング・システムを使用するフェデレーテッド・サーバー上にラッパーを xml_wrapper の名前で登録するには、次のステートメントを発行します。

```
CREATE WRAPPER xml_wrapper LIBRARY 'libdb2lsxml.a';
```

指定するラッパー・ライブラリー・ファイルの名前は、フェデレーテッド・サーバーのオペレーティング・システムによって異なります。CREATE WRAPPER ステートメントに指定する正しい名前については、『XML ラッパー・ライブラリー・ファイル』のリストを参照してください。

この一連の作業における次のタスクは、『XML ラッパーのサーバー定義の登録』です。

関連資料:

- 468 ページの『XML ラッパー・ライブラリー・ファイル』
- 「SQL リファレンス 第 2 巻」の『CREATE WRAPPER ステートメント』

XML ラッパー・ライブラリー・ファイル

次の表には、XML ラッパーのディレクトリー・パスおよびライブラリー・ファイル名をリストしています。

DB2 Information Integrator をインストールすると、この表にリストされているディレクトリー・パスに 3 つのライブラリー・ファイルが追加されます。たとえば、フェデレーテッド・サーバーが AIX 上で稼働している場合、ディレクトリー・パスに追加されるラッパー・ライブラリー・ファイルは libdb2lsxml.a、libdb2lsxmlf.a、および libdb2lsxmlu.a です。

ラッパーを登録する場合、表にリストされているライブラリー・ファイル名だけを指定してください。

表 91. XML ラッパー・ライブラリーのロケーションとファイル名

オペレーティング・システム	ディレクトリー・パス	ラッパー・ライブラリー・ファイル
AIX	/usr/opt/db2_08_01/lib/	libdb2lsxml.a
HP-UX	/opt/IBM/db2/V8.1/lib	libdb2lsxml.sl
Linux	/opt/IBM/db2/V8.1/lib	libdb2lsxml.so
Solaris	/opt/IBM/db2/V8.1/lib	libdb2lsxml.so
Windows	%DB2PATH%¥bin	db2lsxml.dll

| %DB2PATH% は、Windows にインストールされている DB2 Information Integrator
| のディレクトリーを指定するのに使用する環境変数です。デフォルトの Windows
| ディレクトリー・パスは C:\Program Files\IBM\SQLLIB です。

| **関連タスク:**

- | • 467 ページの『XML ラッパーの登録』

XML データ・ソースのサーバーの登録

XML データ・ソースのサーバーの登録は、XML をフェデレーテッド・システム
に追加するという、より大きな作業の一部を成しています。ラッパーを登録した
後、対応するサーバーを登録する必要があります。

制約事項:

XML ラッパーは TYPE および VERSION キーワードを使用しません。CREATE
SERVER ステートメントにこれらのキーワードを使用するとエラーが生じます。

XML ラッパーは、フェデレーテッド・システムに対するパススルー・セッションを
サポートしません。

手順:

XML サーバーをフェデレーテッド・システムに登録するには、CREATE SERVER
ステートメントを発行します。

例:

```
CREATE SERVER xml_server WRAPPER my_xml;
```

この一連の作業における次のタスクは、『XML データ・ソースのニックネームの登
録』です。

プロキシ・サーバーを介して XML 文書にアクセスする

プロキシ・サーバーを介して XML 文書にアクセスするには、サーバー定義を作
成するときにオプションを指定する必要があります。指定するオプションは、アク
セスしたいプロキシ・サーバーのタイプによって異なります。

以下の条件のすべてが真である場合は、CREATE SERVER ステートメントでプロキ
シー・オプションを使用する必要があります。

- URI を使用してデータを検索したい。
- 使用する URI はプロキシを介してファイアウォールの反対側にあるデータを
検索する。
- 使用されるファイアウォールまたはプロキシは、HTTP、SOCKS4、または
SOCKS5 である。

使用するプロキシのタイプ、およびプロキシ・オプションに指定する必要のあ
る設定については、ネットワーク管理者に問い合わせてください。

HTTP プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、HTTP プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_h
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'HTTP', PROXY_SERVER_NAME 'proxy_h',
          PROXY_SERVER_PORT '8080');
```

SOCKS4 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、SOCKS4 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_s4
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS4', PROXY_SERVER_NAME 'proxy_4',
          PROXY_SERVER_PORT '1080');
```

認証情報のない SOCKS5 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、認証情報のない SOCKS5 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_s5
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS5', PROXY_SERVER_NAME 'proxy_5',
          PROXY_SERVER_PORT '1081');
```

認証情報のある SOCKS5 プロキシ・サーバーのサーバー定義を登録する例:

サーバー定義を登録して、認証情報のある SOCKS5 プロキシ・サーバーを指定するには、以下のステートメントを使用します。

```
CREATE SERVER xml_server_s5a
  WRAPPER xml_wrapper
  OPTIONS (PROXY_TYPE 'SOCKS5', PROXY_SERVER_NAME 'proxy_5',
          PROXY_SERVER_PORT '1081', PROXY_AUTHID 'Martin',
          PROXY_PASSWORD 'aaa', );
```

XML 検証機能は、プロキシ機能と共に使用するとき、いくつかの制限事項が生じることがあります。この制限事項が生じる条件は、以下のとおりです。

- さまざまなプロキシ・オプションを設定したプロキシ機能を、サーバー・レベルで使用している。
- XML インスタンス文書に、ファイアウォールの外側に存在する外部 XML スキーマへの参照が含まれる。

これらの条件のいずれかが適合する場合、XML スキーマのロケーションをファイアウォールの内側に変更してみてください。XML スキーマ・ロケーションを変更する場合、XML インスタンス文書を XML スキーマの新しいロケーションに合わせて更新する必要があります。

関連タスク:

- 474 ページの『XML データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

XML データ・ソースのニックネームの登録

ニックネームと XML 文書間のデータ関連

ニックネームは、XML 文書データのツリー構造に対応しています。親ニックネームと子ニックネームは、データ・ツリー構造のルート構造とネスト・エレメントに対応します。それら親ニックネームと子ニックネームは、CREATE NICKNAME ステートメントで指定される主キーおよび外部キーによって結び付けられます。

各ニックネームは、以下の機能を実行する XPath 式によって定義されます。

- 個々のタプルを表す XML エレメントを識別する。
- 各エレメントから列値を取り出す方法を指定する。

XML ラッパーは、XPath 式を使用することによって、XML 文書のデータと、リレーショナル表の行との間の対応を確立します。それらの XPath 式は、XML 文書内の値を特定し、それらの値が各行の列にどう対応するかを決定します。XML ラッパーは、XML 文書データを読むだけです。XML ラッパーは、そのデータを更新しません。

ニックネームを作成する際、そのニックネームと XML 文書との間の関連付けを指定するためのオプションを選択します。ニックネームは、固定した方法で XML 文書に関連付けられるか、または指定するソース名によって XML 文書に関連付けられます。

固定した関連付けの場合、ニックネームは特定の XML 文書のデータを表します。それらの XML 文書には、次のものが含まれます。

1 つのローカル・ファイル

XML 文書として 1 つの XML ファイルを指定します。

1 つのディレクトリー・パスに含まれる複数のローカル・ファイル

複数の XML ファイルの存在するディレクトリー・パスを指定します。ニックネームには、そのディレクトリー・パス内の XML ファイルから XML 文書データが提供されます。それらの XML ファイルの構成は、すべて同じでなければなりません。ニックネームの構成とは違う構成の XML ファイルがそのディレクトリーに含まれていると、XML ラッパーがその XML データ・ファイルを処理した時点で NULL 値が戻されます。ディレクトリーは、フェデレーテッド・サーバーのローカル・ディレクトリーか、または共有ファイル・システムからアクセス可能なディレクトリーでなければなりません。

注: XML ラッパーがディレクトリーをスキャンする際には、拡張子が .xml のファイルのみが採用および構文解析されます。その他のファイル (拡張子が .txt や .xsd のファイル、あるいは拡張のないファイルなど) は、XML ラッパーによりすべて無視されます。

ファイルの固定データを指定するには、CREATE NICKNAME ステートメントの FILE_PATH オプションを使用します。ディレクトリーからの固定データを指定するには、DIRECTORY_PATH オプションを使用します。

照会実行中にソース・データを指定する場合には、スキーマがニックネーム定義に一致する任意の XML 文書ソースからのデータを表すニックネームを使用できます。それらの XML 文書には、次のものが含まれます。

URI (Uniform Reference Identifier)

URI によって参照されるリモート XML ファイルからの XML 文書データが、ニックネームに提供されます。(この文書ソースを指定するには DOCUMENT 'URI' ニックネーム列オプションを使用します。)

リレーショナル列

XML 文書への入力としてリレーショナル表、ビュー、またはニックネームの列が使用されます。(この文書ソースを指定するには DOCUMENT 'COLUMN' ニックネーム列オプションを使用します。)

ファイル

照会実行時に、XML データが含まれる単一のファイルが入力として提供されます。(この文書ソースを指定するには DOCUMENT 'FILE' ニックネーム列オプションを使用します。)

ディレクトリー

照会実行時に、指定したディレクトリー・パスに含まれる複数の XML ファイルからデータが提供されます。(この文書ソースを指定するには DOCUMENT 'DIRECTORY' ニックネーム列オプションを使用します。)

照会時にソース・データが提供されることを指定するには、DOCUMENT 列オプションを指定します。DOCUMENT 列に URI、COLUMN、FILE、または DIRECTORY のいずれかを指定することにより、XML 文書ソースの種類を指定します。

DOCUMENT 列オプションと共に、FILE_PATH オプションまたは DIRECTORY_PATH オプションを指定することはできません。

使用するデータの種類には関係なく (固定形式のデータでも、照会時に指定されるソース名から得られるデータでも)、STREAMING オプションを指定すれば、XML ラッパーは XML 文書データを各フラグメントに分割します。XML ラッパーは、その結果得られる XML データのストリームを処理し、照会フラグメントによって要求されている情報を抽出します。XML ラッパーは、フラグメントを一度に 1 つずつ解析します。フラグメントが一度に 1 つずつ解析されることにより合計メモリーは少なくなりますが、照会全体の実行にかかる処理時間はサーバーのメモリー容量に応じて増加します。したがって、STREAMING オプションを使用するのは、大規模な XML 文書を解析する場合だけにしてください (50 MB 以上の文書の場合)。

さらに、ニックネーム・オプション値を選択すれば、大量の XML データまたは複数のネスト・エレメントを含むデータを取り出す照会を最適化するのに役立ちます。それらのオプションには、次のものが含まれます。

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME
- NEXT_TIME

これらのオプションの値をいろいろと設定してみるにより、XML 照会をテストし、最適化することができます。それらのオプション値によって、エレメントを検出して XML の行に含まれるデータを解析するのに必要な処理時間が制御されません。

関連概念:

- 463 ページの『XML とは?』
- 473 ページの『XML ラッパーのコスト・モデル機能』
- 480 ページの『XML コスト・モデル機能の最適化のヒント』

関連タスク:

- 474 ページの『XML データ・ソースのニックネームの登録』

関連資料:

- 610 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

XML ラッパーのコスト・モデル機能

XML ラッパーには、XML ソース文書に対応するニックネームに対する照会を最適化するためのコスト・モデル機能が含まれています。

CREATE NICKNAME ステートメントを使ってニックネームを作成する際には、コスト・モデル機能をサポートするために、ニックネーム・オプション値として次のパラメーターを指定できます。

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

これらのパラメーターには、デフォルト値を使用できます。あるいは、これらのパラメーターに値を指定することによって、作成するルート・ニックネームおよびルート以外のニックネームに対する照会を最適化することもできます。

INSTANCE_PARSE_TIME パラメーターは、ルート・ニックネームのエレメントのうち行を生成するルート・エレメント (例: customer) を 1 つ読んで構文解析するのに必要な時間 (ミリ秒) です。それには、その中に含まれていて行を生成することになる非ルート・エレメント (例: 各 customer の order、payment、および item) も含まれます。XML ラッパーは、それら行生成ルート・エレメントおよび行生成非ルート・エレメントを表す構造をメモリー中にビルドします。

XPATH_EVAL_TIME パラメーターは、ニックネームの行に対応するデータを探索するための XPath 式を評価するのに必要な時間 (ミリ秒) です。評価される XPath 式には、実際の行を探索するための XPath 式に加えて、それらの行の中に含まれる列値を探索する XPath 式も含まれます。

関連概念:

- 463 ページの『XML とは?』
- 471 ページの『ニックネームと XML 文書間のデータ関連』
- 480 ページの『XML コスト・モデル機能の最適化のヒント』

関連資料:

- 610 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

XML データ・ソースのニックネームの登録

XML データ・ソースのニックネームの登録は、XML をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。XML データ・ソースのツリー構造に対応するニックネームを作成する必要があります。親ニックネームは、ツリーのルート構造に対応します。子ニックネームは、親ニックネームの要素内でネストされている要素に対応します。

前提条件:

データベースのコード・ページは、XML ソース・ファイルの文字セットと一致しなければなりません。

制約事項:

ネーム・スペースはサポートされていません。

手順:

XML データ・ソースのニックネームを登録するには、CREATE NICKNAME ステートメントを発行します。

この一連の作業における次のタスクは、『非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)』です。

関連概念:

- 471 ページの『ニックネームと XML 文書間のデータ関連』
- 473 ページの『XML ラッパーのコスト・モデル機能』

関連タスク:

- 466 ページの『フェデレーテッド・システムへの XML の追加』
- 479 ページの『非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)』
- 74 ページの『非リレーショナル・データ・ソースのニックネーム列の指定』

関連資料:

- 610 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

CREATE NICKNAME ステートメント - XML ラッパーの例

このトピックでは、CREATE NICKNAME ステートメントを使用して XML ラッパーのニックネームを登録する方法を示すいくつかの例が提供されます。このトピックには、親と子のニックネームを作成する完全指定の例、特定の列オプションの例、およびビューの使用を示す例が含まれます。

| 推奨: 照会内に XPATH 列およびニックネーム・オプションを指定するときには、
| 自己または下位演算子 // を使用しないでください。自己または下位演算子は
| XPath 演算子であり、それを使用するとフェデレーテッド・サーバーのパフォーマ
| ンスが低下する場合があります。

完全指定の例:

以下に示すのは、図 44 に示されるサンプル XML ファイルを使用することによっ
て XML データ・ソースのニックネームを作成する方法の例です。

```
<?xml version="1.0" encoding="UTF-8"?>
      <doc>
        <customer id='123'>
          <name>...</name>
          <address>...</address>
          ...
          <order>
            <amount>...</amount>
            <date>...</date>
            <item quant='12'>
              <name>...</name>
            </item>
            <item quant='4'>...</item>
            ...
          </order>
          <order>...</order>
          ...
          <payment>
            <number>...</number>
            <date>...</date>
          </payment>
          <payment>...</payment>
          ...
        </customer>
        <customer id='124'>...</customer>
      </doc>
```

図 44. サンプル XML ファイル

親ニックネーム:

最初のステップは、親ニックネーム customers の作成です。ニックネームを作成す
るには、次のステートメントを発行します。

```
CREATE NICKNAME customers
(
  id          VARCHAR(5)  OPTIONS(XPATH './@id')
  name       VARCHAR(16)  OPTIONS(XPATH './name'),
  address    VARCHAR(30)  OPTIONS(XPATH './address/@street'),
  cid       VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(DIRECTORY_PATH '/home/db2user',
        XPATH '/doc/customer', STREAMING 'YES');
```

このステートメントは、指定されたディレクトリー・パス /home/db2user のもとに
ある複数の XML ファイルに関する customers ニックネームを作成します。
STREAMING ニックネーム・オプションは、XML ソース・データがノード (この
例では、カスタマー・レコード) ごとに分けられて処理されるということを示しま
す。STREAMING ニックネーム・オプションが使用されている場合は、ラッパーは

XML 文書全体をメモリーに保管することはしません。その代わりに、XML ラッパーは文書を順次個別に構文解析できる複数のセクションに分割します。
STREAMING ニックネーム・オプションは、サイズの大きい XML 文書にのみ使用してください。このオプションを使用すると、照会のパフォーマンスは影響を受けます。

子ニックネーム:

次のステップは、orders、payments、および items の子ニックネームを作成することです。

以下のステートメントを発行して orders 子ニックネームを作成します。

```
CREATE NICKNAME orders
(
  amount  INTEGER      OPTIONS(XPATH './amount'),
  date    VARCHAR(10)  OPTIONS(XPATH './date'),
  oid     VARCHAR(16)  OPTIONS(PRIMARY_KEY 'YES'),
  cid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './order');
```

以下のステートメントを発行して payments 子ニックネームを作成します。

```
CREATE NICKNAME payments
(
  number  INTEGER      OPTIONS(XPATH './number'),
  date    VARCHAR(10)  OPTIONS(XPATH './date'),
  cid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'CUSTOMERS'))
FOR SERVER xml_server
OPTIONS( XPATH './payment');
```

以下のステートメントを発行して items 子ニックネームを作成します。

```
CREATE NICKNAME items
(
  name     VARCHAR(20)  OPTIONS(XPATH './name'),
  quantity INTEGER      OPTIONS(XPATH './@quant'),
  oid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(FOREIGN_KEY 'ORDERS'))
FOR SERVER xml_server
OPTIONS( XPATH './item');
```

列オプションの例:

以下の例は、ニックネームを作成するときに DOCUMENT 列オプションを組み込む方法を示したものです。この例には、それらのオプションを照会内で使用する方法も示されています。

以下の CREATE NICKNAME の例は、DOCUMENT 'FILE' 列オプションの使用例を示しています。

```
CREATE NICKNAME customers
(
  doc     VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name    VARCHAR(16)   OPTIONS(XPATH './name'),
  address VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16)  FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH '/doc/customer');
```

以下の照会は (WHERE 文節で XML 文書のロケーションを指定している) customers ニックネームで実行することができるようになります。

```
SELECT * FROM customers WHERE doc = '/home/db2user/Customers.xml';
```

以下の CREATE NICKNAME の例は、DOCUMENT 'DIRECTORY' 列オプションの使用例を示しています。

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'DIRECTORY'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH '/doc/customer');
```

このとき、以下の照会を customers ニックネームに対して実行できます。

```
SELECT name FROM customers WHERE doc = '/home/data/xml';
```

この照会は、WHERE 文節で指定されるディレクトリー・パス /home/data/xml のもとに位置指定される XML 文書を探索します。

以下の CREATE NICKNAME の例は、DOCUMENT 'URI' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'URI'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid      VARCHAR(16)   FOR BIT DATA NOT NULL OPTIONS(PRIMARY_KEY 'YES'))
FOR SERVER xml_server
OPTIONS(XPATH '/doc/customer');
```

このとき、以下の照会を customers ニックネームに対して実行し、XML データをリモート・ロケーションから検索できます。

```
SELECT * FROM customers WHERE doc = 'http://www.lg-mv.org/foo.xml';
```

以下の CREATE NICKNAME の例は、DOCUMENT 'COLUMN' ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME emp
(
  doc      VARCHAR(500)  OPTIONS(DOCUMENT 'COLUMN')
  fname    VARCHAR(16)   OPTIONS(XPATH '@first'),
  lname    VARCHAR(16)   OPTIONS(XPATH '@last'))
FOR SERVER xml_server
OPTIONS(XPATH '/doc/name');
```

このとき、以下の照会の 1 つを emp ニックネームに対して実行し、XML データを検索できます。

```
SELECT * FROM emp WHERE doc = '<?xml version="1.0" encoding="UTF-8"?>
<doc>
  <title> employees </title>
  <name first="David" last="Marston"/>
  <name first="Donald" last="Leslie"/>
  <name first="Emily" last="Farmer"/>
  <name first="Myriam" last="Midy"/>
```

```
<name first="Lee" last="Tran"/>
<name first="Lili" last="Farmer"/>
<name first="Sanjay" last="Kumar"/>
</doc>';
```

または

```
SELECT * FROM emp WHERE doc = (SELECT * FROM xml_tab);
```

xml_tab 表には、XML データが追加される 1 つの列が含まれます。

ビューの例:

以下の例は、ルートではないニックネームのビューを作成して XML ソース文書を記述する方法を示しています。これらの例では 図 45 に示されているサンプル・ファイルのニックネームは、前に customers、orders、payments、および items として作成されていたものと想定しています。

```
<?xml version="1.0" encoding="UTF-8"?>
  <doc>
    <customer id='123'>
      <name>...</name>
      <address>...</address>
      ...
      <order>
        <amount>...</amount>
        <date>...</date>
        <item quant='12'>
          <name>...</name>
        </item>
        <item quant='4'>...</item>
        ...
      </order>
      <order>...</order>
      ...
      <payment>
        <number>...</number>
        <date>...</date>
      </payment>
      <payment>...</payment>
      ...
    </customer>
    <customer id='124'>...</customer>
  </doc>
```

図 45. サンプル XML ファイル

以下の例は、ルートではないニックネーム order のビューを作成する方法を示しています。

```
CREATE VIEW order_view AS
SELECT o.amount, o.date, o.oid, c.cid
FROM customers c, orders o
WHERE c.cid = o.cid;
```

以下の例は、ルートではないニックネーム payment のビューを作成する方法を示しています。

```
CREATE VIEW payment_view AS
  SELECT p.number, p.date, c.cid
FROM customers c, payments p
WHERE c.cid = p.cid;
```

以下の例は、ルートではないニックネーム `item` のビューを作成する方法を示しています。

```
CREATE VIEW item_view AS
  SELECT i.quantity, i.name, o.oid
FROM customers c, orders o, items i
WHERE c.cid = o.cid AND o.oid = i.oid;
```

これらのビューにサブミットされた照会は、ルート・ディレクトリーへの結合パスが存在するので、正しく処理されます。

たとえば、以下の照会は、カスタマーのオーダーの金額と、同じ日付からの支払いをペアにします。

```
SELECT o.amount, p.amount
FROM order_view o, payment_view p
WHERE p.date = o.date AND
      p.cid = o.cid;
```

関連タスク:

- 474 ページの『XML データ・ソースのニックネームの登録』

関連資料:

- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』
- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』
- 610 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』

非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー)

非ルート・ニックネームのフェデレーテッド・ビューの作成 (XML ラッパー) は、XML をフェデレーテッド・システムに追加するという、より大きな作業の一部を成しています。

XML 文書を説明するニックネームの階層に対するフェデレーテッド・ビューを定義できます。フェデレーテッド・ビューを定義することにより、XML ニックネーム階層 (ルート・ニックネーム、および特別な `PRIMARY_KEY` と `FOREIGN_KEY` 列以外の列を結合する照会は含まれない) の部分を結合する照会が、正しく実行されることが保証されます。

手順:

必要なすべての述部およびルート・ディレクトリーへの絶対パスを含むフェデレーテッド・ビューを定義するには、以下のステップを実行します。

1. 各非ルート・ニックネームのビューを、ルートへのパス上のすべてのニックネームの結合として定義します。

2. WHERE 文節において、PRIMARY_KEY および FOREIGN_KEY 列に対する結合述部を作成します。
3. SELECT リストに、非ルート・ニックネームのすべての列を組み込みます。ただし、FOREIGN_KEY ニックネーム列オプションによって指定された列は除きます。
4. SELECT リストに、PRIMARY_KEY オプションによって指定された親ニックネームの列を組み込みます。

関連資料:

- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

XML コスト・モデル機能の最適化のヒント

XML ラッパーのコスト・モデル機能は、作成するニックネームに対する照会を最適化するのに役立ちます。

コスト・モデル機能では、CREATE NICKNAME ステートメントの次のパラメーターを使用します。

- INSTANCE_PARSE_TIME
- XPATH_EVAL_TIME

これらのパラメーターの値は、CREATE NICKNAME ステートメントを発行して XML データ・ソースのニックネームを登録する際に指定できます。

コスト・モデル機能は、XML ソース文書の各行データを解析し、ニックネームの XPath 式を評価するのに必要な時間を調べる際に、それらのパラメーター値を使用します。

これらのパラメーターには、デフォルト値を使用できます。しかし、作成するニックネームの大規模なまたは複雑な XML ソース構造に関する照会を最適化する場合には、次の例に示されている方針に従ってください。

大規模照会を最適化する例:

XML 文書に、次の 4 つのニックネームを含むリレーショナル・スキーマが含まれているとします。

- customers
- orders
- payments
- items

また、ニックネーム customers がルート・ニックネームであるとしています。

各ニックネームに対して照会を実行します。実際の環境でよくある XML データのサンプルに対して、各照会を実行します。

例:

```
SELECT * from customers;
SELECT * from orders;
SELECT * from payments;
SELECT * from items;
```

db2batch コマンドか、それに相当するコマンドやユーティリティーを使用することによって、各照会の実行に必要な時間 (ミリ秒) を調べます。(db2batch コマンドを使うと、照会実行に必要な時間を内容として含む出力ファイルが生成されます。) また、戻されるタプルの数も記録します。

各ニックネームごとに、次の式により **INSTANCE_PARSE_TIME** パラメーターと **XPATH_EVAL_TIME** パラメーターに最適な値を決定します。

```
INSTANCE_PARSE_TIME = (75% × [SELECT * query の実行時間]) ÷ [戻されるタプルの数]
XPATH_EVAL_TIME     = (25% × [SELECT * query の実行時間]) ÷ [戻されるタプルの数]
```

ルート・ニックネーム (この例では **customers** です) については、**INSTANCE_PARSE_TIME** パラメーターと **XPATH_EVAL_TIME** パラメーターに対して、上記の式で計算される値を使用します。

非ルート・ニックネーム (この例では **orders**、**payments**、および **items** です) については、**XPATH_EVAL_TIME** パラメーターについてのみ、上記の式で計算される値を使用します。**INSTANCE_PARSE_TIME** パラメーターの値は、非ルート・ニックネームには適用されません。

これらの式は、照会を調整するための 1 つの方針として使用できます。これらのパラメーターに最適な値は、XML ソース文書の複雑さや、使用しているプロセッサの速度に応じて異なります。

関連概念:

- 463 ページの『XML とは?』
- 471 ページの『ニックネームと XML 文書の間データ関連』
- 473 ページの『XML ラッパーのコスト・モデル機能』

関連資料:

- 「コマンド・リファレンス」の『db2batch - ベンチマーク・ツール・コマンド』

XML データ・ソース - 照会の例

このトピックでは、ニックネーム **customers**、**orders**、および **items** を使用するいくつかのサンプル照会を提供します。これらのニックネームは、**CREATE NICKNAME** ステートメントを使用して以前に登録されました。

以下の照会は、すべてのカスタマー名を表示します。

```
SELECT name FROM customers;
```

以下の照会は、カスタマー名が **Chang** のすべてのレコードを表示します。

```
SELECT * FROM customers
WHERE name='Chang';
```

以下の照会は、各カスタマーの各オーダーごとのカスタマー名を表示します。

```
SELECT c.name, o.amount FROM customers c, orders o
WHERE c.cid=o.cid;
```

customers ニックネームおよび orders ニックネームの間の親子関係を示すには、結合 c.cid=o.cid を指定しなければなりません。

以下の照会は、オーダーごとおよび各カスタマーの品目ごとの、カスタマーの住所、オーダー金額、および品目名を選択します。

```
SELECT c.address, o.amount, i.name FROM customers c, orders o, items i
WHERE c.cid=o.cid AND o.oid=i.oid;
```

親子関係を維持するには、これら 2 つの結合を指定する必要があります。

以下の例は、FILE_PATH ニックネーム・オプションは指定しないが、DOCUMENT 列オプションは指定するニックネームを使用して、照会を作成する方法を示しています。 customers ニックネームを作成するために使用される、対応する CREATE NICKNAME ステートメントが、以下に示されています。

```
CREATE NICKNAME customers
(
  doc      VARCHAR(100)  OPTIONS(DOCUMENT 'FILE'),
  name     VARCHAR(16)   OPTIONS(XPATH './name'),
  address  VARCHAR(30)   OPTIONS(XPATH './address/@street'),
  cid     VARCHAR(16)   OPTIONS(PRIMARY_KEY 'YES')
FOR SERVER xml_server
OPTIONS(XPATH '/doc/customer');
```

以下の照会は、/home/db2user/Customers.xml のファイル・パスをもつ XML ファイル Customers.xml からすべてのデータを選択します。

```
SELECT * FROM customers
WHERE doc='/home/db2user/Customers.xml';
```

以下の照会は、Customers.xml ファイルから、1000 の金額を超えるオーダーについて、カスタマーの名前とオーダーの日付を選択します。

```
SELECT c.name, o.date FROM customers c, orders o
WHERE c.doc='/home/db2user/Customers.xml' AND o.amount > 1000;
```

ファイル・パス /home/db2user/Customers.xml は、Customers.xml ファイルのロケーションを指定します。

関連資料:

- 610 ページの『CREATE NICKNAME ステートメント構文 - XML ラッパー』
- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

XML ラッパーのメッセージ

ここでは、XML のラッパーを処理するときに出されるメッセージについて説明します。メッセージの詳細については、「DB2 メッセージ・リファレンス」を参照してください。

表 92. XML のラッパーによって出されるメッセージ

エラー・コード	メッセージ	説明
SQL0405N	数値リテラル "<column_name>" は、値が有効な値の範囲外になっているため、無効です。	示された数値リテラルが適切な範囲内にありません。 CREATE NICKNAME ステートメント内の列のデータ・タイプをチェックしてください。
SQL0408N	値には、その割り当てターゲットのデータ・タイプとの互換性がありません。ターゲット名は "<column_name>" です。	列に割り当てられる値のデータ・タイプが、割り当てターゲットの宣言されたデータ・タイプと互換性がありません。 CREATE NICKNAME ステートメント内の列のデータ・タイプをチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「ラッパー・オブジェクトの作成エラー」。)	新しいラッパー・オブジェクトを作成しているときにエラーが発生しました。 IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「<xerces_xalan_error_message>」。)	Xerces または Xalan の関数呼び出し中にエラーが発生しました。 XML 文書をチェックしてください。文書が適切に構造化されている場合、エラー・メッセージの詳細については Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XalanDOMException: 例外コードは <exception_code>」。)	XalanDOMException 例外が発生しました。例外コードの詳細については、Xalan の資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XMLException: <exception_error_message>」。)	XMLException 例外が発生しました。例外コードの詳細については、Xalan の資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XSLEException: <exception_error_message>」。)	XSLEException 例外が発生しました。例外コードの詳細については、Xalan の資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「SAXParseException: <exception_error_message>」。)	SAXParseException 例外が発生しました。例外コードの詳細については、Xalan の資料を参照してください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「ノード値の取得エラー」。)	Xalan が無効なノードにアクセスしようとした。IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XML 文書の構文解析エラー」。)	XML 文書を構文解析しているときにエラーが発生しました。XML 文書をチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XML 文書のルート・エレメントの取得エラー」。)	XML 文書を構文解析した後に、Xalan がルート・エレメントを検索しようとしたが失敗しました。XML 文書をチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「XPath 式評価中の未指定の例外」。)	XPath 式の評価中に、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「ノード値取得中の未指定の例外」。)	ノード値の検索中に、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「入力文書構文解析中の未指定の例外」。)	XML 文書の構文解析中に、Xalan によって未指定の例外が生成されました。XML 文書をチェックし、Xalan 資料を参照してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「基数評価時のエラー」。)	IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「<SOAP_error_message>」。)	SOAP ライブラリーからエラーが出されました。SQL ステートメントのエラーを解決できない場合は、IBM ソフトウェア・サポート担当者に連絡してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「無効な URI」。)	ラッパーは指定された URL にアクセスできませんでした。URL がアクセス可能なものかどうかを確認してください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「無効な XML 文書コンテンツ」。)	XML 文書の内容が妥当ではありません。文書が整形形式の構造であることを確認してください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「無効な SOAP エンベロープ」。)	SOAP エンベロープが無効です。その構文と内容をチェックしてください。
SQL0901N	重大ではないシステム・エラーにより、SQL ステートメントが失敗しました。後続の SQL ステートメントは処理できません。(理由「メモリー割り振りエラー」。)	メモリー割り振りの際に、エラーが発生しました。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「正しくない DATE 形式」です。	XML 文書中の日付値の形式が不正です。日付値の有効な形式は yyyy-mm-dd です。XML 文書をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「列データ・タイプがサポートされていない」です。	ニックネーム列がサポートされていないデータ・タイプをもっています。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「TYPE 文節がサポートされていない」です。	CREATE SERVER ステートメントに TYPE 文節が含まれています。この文節は XML ラッパーによってサポートされていません。この文節を削除してください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「VERSION 文節がサポートされていない」です。	CREATE SERVER ステートメントに VERSION 文節が含まれています。この文節は XML ラッパーによってサポートされていません。この文節を削除してください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DOCUMENT 列と共に使用する述部が無効」です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FOREIGN_KEY 列と共に使用する述部が無効」です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「PRIMARY_KEY 列と共に使用する述部が無効」です。	照会に、誤ったオペランドのある述部が含まれています。照会の中の述部をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「XPATH と DOCUMENT オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「XPATH と FOREIGN_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「XPATH と PRIMARY_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DOCUMENT と FOREIGN_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DOCUMENT と PRIMARY_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FOREIGN_KEY と PRIMARY_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「列オプションが欠落」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DOCUMENT 列オプションが固有でない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FOREIGN_KEY 列オプションが固有でない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「PRIMARY_KEY 列オプションが固有でない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「無効な DOCUMENT オプション値」です。	CREATE NICKNAME ステートメントに指定されている DOCUMENT オプションの値が無効です。値は FILE でなければなりません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「無効な PRIMARY_KEY オプション値」です。	CREATE NICKNAME ステートメントに指定されている PRIMARY_KEY オプションの値が無効です。値は YES でなければなりません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「無効な FOREIGN_KEY オプション値」です。	CREATE NICKNAME ステートメントに指定されている FOREIGN_KEY オプションの値が無効です。値が、どの親ニックネームとも一致しません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FILE_PATH と DOCUMENT オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。FILE_PATH および DOCUMENT オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FILE_PATH と SOAP オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。FILE_PATH および SOAP オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DIRECTORY_PATH と SOAP オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。DIRECTORY_PATH および SOAP オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FILE_PATH と DIRECTORY_PATH オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。FILE_PATH および DIRECTORY_PATH オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「VALIDATE と STREAMING オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。VALIDATE および STREAMING オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「FILE_PATH と FOREIGN_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。FILE_PATH および FOREIGN_KEY オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「DIRECTORY_PATH と FOREIGN_KEY オプションに互換性がない」です。	CREATE NICKNAME ステートメントの指定が正しくありません。DIRECTORY_PATH および FOREIGN_KEY オプションは同時に指定することはできません。CREATE NICKNAME ステートメントの構文をチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「STREAMING が使用可能な場合、XPATH オプション値は無効」です。	STREAMING 機能が使用可能になっている場合、ニックネーム XPATH 式は無効です。 XPATH オプションに、 /、 ./、 // など、無効な値が含まれていないかどうかを確認してください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「XML ファイルの読み取り不可」です。	CREATE NICKNAME ステートメントまたは照会の中に指定されているファイル・パスが無効です。指定されたファイルが存在しません。 CREATE NICKNAME ステートメントと照会をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「ディレクトリーのオープン不可」です。	CREATE NICKNAME ステートメントまたは照会の中に指定されているディレクトリー・パスが無効です。指定されたディレクトリーが存在しません。 CREATE NICKNAME ステートメントと照会をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「XML データへの参照の欠落」です。	CREATE NICKNAME ステートメントには、 XML データへの参照が含まれていなければなりません。 CREATE NICKNAME ステートメントをチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「値 'SOAP' の指定された DOCUMENT 列オプションの欠落」です。	CREATE NICKNAME ステートメントの指定が正しくありません。 DOCUMENT オプションの値を調べてください。値は SOAP でなければなりません。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「SOAP オプションの欠落」です。	CREATE NICKNAME ステートメントの指定が正しくありません。 SOAP オプションを指定する必要があります。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「INSTANCE_PARSE_TIME はルート・ニックネームについてのみ」です。	CREATE NICKNAME ステートメントの指定が正しくありません。 INSTANCE_PARSE_TIME 値を指定できるのは、ルート・ニックネームについてのみです。 CREATE NICKNAME 構文をチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「VALIDATE はルート・ニックネームについてのみ」です。	CREATE NICKNAME ステートメントの指定が正しくありません。VALIDATE オプションを YES に設定できるのは、指定したニックネームがルート・ニックネームの場合のみです。CREATE NICKNAME 構文をチェックしてください。
SQL1822N	データ・ソース "XML wrapper" から予期しないエラー・コード "<trace_point>" を受け取りました。関連するテキストおよびトークンは「STREAMING はルート・ニックネームについてのみ」です。	CREATE NICKNAME ステートメントの指定が正しくありません。STREAMING オプションを YES に設定できるのは、指定したニックネームがルート・ニックネームの場合のみです。CREATE NICKNAME 構文をチェックしてください。
SQL1823N	サーバー "<server_name>" からデータ・タイプ "<data_type_name>" に存在するデータ・タイプ・マッピングがありません。	CREATE NICKNAME ステートメントの指定が正しくありません。列のデータ・タイプが無効です。CREATE NICKNAME 構文をチェックしてください。
SQL1881N	"<option_name>" は、"<object_name>" のための有効な "<option_type>" オプションではありません。	指定されたオプションは存在しないか、このデータ・ソースに対しては無効である可能性があります。CREATE NICKNAME ステートメントをチェックしてください。
SQL1881N	"DIRECTORY_PATH" は、"<object_name>" のための有効な "NICKNAME" ではありません。	CREATE NICKNAME ステートメントに指定されている DIRECTORY_PATH オプションの値が無効です。指定するディレクトリは、ルート・ディレクトリでなければなりません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1882N	"nickname" オプション "VALIDATE" は、"<object_name>" に対して "<option_value>" に設定できません。	CREATE NICKNAME ステートメントに指定されている VALIDATE オプションの値が無効です。この値は、YES または NO でなければなりません。CREATE NICKNAME ステートメントをチェックしてください。

表 92. XML のラッパーによって出されるメッセージ (続き)

エラー・コード	メッセージ	説明
SQL1882N	"nickname" オプション "STREAMING" は、"<object_name>" を "<option_value>" に設定できません。	CREATE NICKNAME ステートメントに指定されている STREAMING オプションの値が無効です。この値は、YES または NO でなければなりません。CREATE NICKNAME ステートメントをチェックしてください。
SQL1883N	"<option_name>" は、"<object_name>" に必須の "<option_type>" オプションです。	必要な DB2 オプションが指定されていませんでした。CREATE NICKNAME ステートメントをチェックしてください。

関連概念:

- 「メッセージ・リファレンス 第 1 巻」の『メッセージの概要』

関連資料:

- 「メッセージ・リファレンス 第 2 巻」の『SQLSTATE メッセージ』

第 4 部 ユーザー定義関数

第 25 章 ライフ・サイエンス・ユーザー定義関数

この章では、ライフ・サイエンス・ユーザー定義関数の概要、それらのフェデレーテッド・システムへの追加方法、そして照会における使用方法について説明します。

ライフ・サイエンス・ユーザー定義関数の概要

ライフ・サイエンス・ユーザー定義関数の概要

ライフ・サイエンス・ユーザー定義関数は、一般にデータ分析に使用されるアルゴリズムを提供します。

ライフ・サイエンス・ユーザー定義関数は、標準の 1 文字コード、および IUPAC-IUB あいまいコードを用いてアミノ酸およびヌクレオチドを示します。

ライフ・サイエンス・ユーザー定義関数は、非リレーショナル・ラッパーのライフ・サイエンス・ユーザー定義関数コンポーネントを使用してインストールします。ライフ・サイエンス・ユーザー定義関数をインストールした後で、それらの関数を登録する必要があります。

ネーム・スペースの競合を回避するために、すべてのライフ・サイエンス・ユーザー定義関数は DB2LS スキーマに登録されます。

関連概念:

- ・ 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator の非リレーショナル・ラッパー』
- ・ 「*IBM DB2 Information Integrator インストール・ガイド*」の『DB2 Information Integrator のリレーショナル・ラッパー』

関連タスク:

- ・ 497 ページの『ライフ・サイエンス・ユーザー定義関数の登録』
- ・ 499 ページの『ライフ・サイエンス・ユーザー定義関数の使用不可』

関連資料:

- ・ 495 ページの『ライフ・サイエンス・ユーザー定義関数ライブラリー・ファイル』
- ・ 496 ページの『機能カテゴリーごとのライフ・サイエンス・ユーザー定義関数』

ライフ・サイエンス・ユーザー定義関数ライブラリー・ファイル

DB2 Information Integrator に付属するユーザー定義関数の中には、ライブラリー・ファイルが必要とするものがあります。これらのライブラリー・ファイルは、フェデレーテッド・サーバー上でユーザー定義関数を登録するときに必要です。

非リレーショナル・データ・ソースに対して DB2 Information Integrator をインストールする際、フェデレーテッド・サーバーに次のユーザー定義ライブラリー・ファイルがインストールされます。

ライフ・サイエンス・ユーザー定義関数ライブラリー:

表 93. ライフ・サイエンスユーザー定義関数ライブラリーのロケーションとファイル名

関数タイプ	オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
ライフ・サイエンス・ユーザー定義関数	AIX	/SQLLIB/function	libdb2lsudfs.a
ライフ・サイエンス・ユーザー定義関数	HP-UX	/SQLLIB/function	libdb2lsudfs.sl
ライフ・サイエンス・ユーザー定義関数	Linux	/SQLLIB/function	libdb2lsudfs.so
ライフ・サイエンス・ユーザー定義関数	Solaris	/SQLLIB/function	libdb2lsudfs.so
ライフ・サイエンス・ユーザー定義関数	Windows	%DB2PATH%\%bin	db2lsudfs.dll

デフォルトの Windows ディレクトリー・パスは、C:\Program Files\IBM\SQLLIB です。 %DB2PATH% は、Windows にインストールされている DB2 Information Integrator のディレクトリーを指定するのに使用する環境変数です。

LSGeneWise ユーザー定義関数ライブラリー:

LSGeneWise ユーザー定義関数は、ライブラリーを別途必要とします。

表 94. LSGeneWise 関数ライブラリーのロケーションとファイル名

関数タイプ	オペレーティング・システム	ディレクトリー・パス	ライブラリー・ファイル名
LSGeneWise 関数	UNIX	/SQLLIB/lib	libdb2lsSTgenewise.a

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

機能カテゴリーごとのライフ・サイエンス・ユーザー定義関数

表 95 に、ライフ・サイエンス・ユーザー定義関数を機能カテゴリーごとにリストし、各カテゴリーの要旨を示します。

表 95. ライフ・サイエンス・ユーザー定義関数

機能カテゴリー	ユーザー定義関数	説明
逆変換	LSPep2AmbNuc、 LSPep2ProbNuc	アミノ酸シーケンスをヌクレオチド・シーケンスに変換します。

表 95. ライフ・サイエンス・ユーザー定義関数 (続き)

機能カテゴリー	ユーザー定義関数	説明
定義行構文解析	LSDefineParse	BLAST ラッパーにより戻される定義行エレメント、または FASTA 形式のデータ・ファイルにある定義行エレメントの構文を解析します。
一般化されたパターン・マッチング	LSPatternMatch、 LSPrositePattern	ヌクレオチド・シーケンスまたはペプチド・シーケンスなどの所定ストリングの特定箇所を識別します。
GeneWise	LSGeneWise	プロテイン・シーケンスをゲノム・シーケンスに合わせます。
Motifs	LSMultiMatch、 LSMultiMatch3、 LSBarCode	ヌクレオチド・シーケンスまたはアミノ酸シーケンスのパターンの一致を図ります。
反転	LSRevNuc、LSRevPep、 LSRevComp	ヌクレオチド・シーケンスまたはアミノ酸シーケンスを反転します。
変換	LSNuc2Pep、 LSTransAllFrames	ヌクレオチド・シーケンスをペプチド・シーケンスに変換します。

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

関連タスク:

- 497 ページの『ライフ・サイエンス・ユーザー定義関数の登録』
- 499 ページの『ライフ・サイエンス・ユーザー定義関数の使用不可』

ライフ・サイエンス・ユーザー定義関数の登録

ライフ・サイエンス・ユーザー定義関数を使用するには、それらの関数を登録する必要があります。

前提条件:

非リレーショナル・ラッパーのライフ・サイエンス・ユーザー定義関数コンポーネントは、DB2 Information Integrator を使用してインストールする必要があります。

手順:

ライフ・サイエンス・ユーザー定義関数を登録するには、enable_LSFunctions コマンドを使用します。

- Windows NT で稼働するフェデレーテッド・サーバーの場合、このコマンドは `sqllib\bin` ディレクトリーにあります。
- AIX で稼働するフェデレーテッド・サーバーの場合、このコマンドは `sqllib/bin` ディレクトリーにあります。

enable_LSFuctions コマンドの構文は以下のとおりです。

```
enable_LSFuctions -n dbName -u userID -p password [-force]
```

dbName

関数の登録先であるフェデレーテッド・データベースの名前。

userID

フェデレーテッド・データベース用の有効なユーザー ID。

password

ユーザー ID の有効なパスワード。

force 関数の除去と再登録に使用するフラグ。このフラグは、誤って破損またはドロップした関数の再登録に使用します。

enable_LSFuctions コマンドは、すべてのライフ・サイエンス・ユーザー定義関数をフェデレーテッド・データベースに登録します。これらの関数は、DB2LS というスキーマ名で登録されます。

enable_LSFuctions コマンドを発行したときに戻される出力例は以下のとおりです。

```
C:> enable_LSFuctions -n federateddb -u db2admin -p db2admin
```

```
(0) Life Sciences Functions were found
-- Create Life Sciences Functions ...
Create Life Sciences Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

関数が登録されており、強制フラグを使用した場合、enable_LSFuctions コマンドを発行したときに戻される出力は以下のようになります。

```
C:> enable_LSFuctions -n federateddb -u db2admin -p db2admin -force
```

```
(21) Life Sciences Functions were found

Life Sciences functions already exist ...
Reinstall Life Sciences functions ...
-- Drop Life Sciences Functions ...
Drop Life Sciences Functions Successfully.
-- Create Life Sciences Functions ...
Create Life Sciences Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

関連タスク:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『DB2 Information Integrator システムへのリレーショナル・ラッパー、非リレーショナル・ラッパー、およびユーザー定義関数の追加』
- 499 ページの『ライフ・サイエンス・ユーザー定義関数の使用不可』

関連資料:

- 496 ページの『機能カテゴリーごとのライフ・サイエンス・ユーザー定義関数』

ライフ・サイエンス・ユーザー定義関数の使用不可

ライフ・サイエンス・ユーザー定義関数を今後使用しない場合は、一時的にその関数を使用不可にするか、またはフェデレーテッド・データベースから永久に関数を除去することができます。

手順:

ライフ・サイエンス・ユーザー定義関数を使用不可にするには、**disable_LSFuctions** コマンドを使用します。

- Windows NT で稼働するフェデレーテッド・サーバーの場合、このコマンドは `sqllib\bin` ディレクトリーにあります。
- AIX で稼働するフェデレーテッド・サーバーの場合、このコマンドは `sqllib/bin` ディレクトリーにあります。

disable_LSFuctions コマンドの構文は以下のとおりです。

```
disable_LSFuctions -n dbName -u userID -p password
```

dbName

関数を使用不可にするフェデレーテッド・データベースの名前。

userID

フェデレーテッド・データベース用の有効なユーザー ID。

password

ユーザー ID の有効なパスワード。

ライフ・サイエンス・ユーザー定義関数の使用不可の例:

disable_LSFuctions コマンドを発行したときに戻される出力例は以下のとおりです。

```
C:>disable_LSFuctions -n federateddb -u db2admin -p db2admin
```

```
(21) Life Sciences Functions were found
-- Drop Life Sciences Functions ...
Drop Life Sciences Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

関数をフェデレーテッド・データベースから恒久的に削除する場合は、関数をアンインストールする必要があります。

関連タスク:

- 「*IBM DB2 Information Integrator インストール・ガイド*」の『リレーショナル・ラッパー、非リレーショナル・ラッパー、およびライフ・サイエンス・ユーザー定義関数の除去 (Windows)』
- 「*IBM DB2 Information Integrator インストール・ガイド*」の『ラッパー、ユーザー定義関数、およびラッパー開発キットの除去 (UNIX)』
- 497 ページの『ライフ・サイエンス・ユーザー定義関数の登録』

逆変換ユーザー定義関数

逆変換ユーザー定義関数 - 概要

逆変換ユーザー定義関数を使用して、ペプチド・シーケンスをヌクレオチド・シーケンスに変換できます。逆変換では変換の逆の処理を行います。

アミノ酸からヌクレオチド・トリプレット・コドンへのマッピングは一对多数であるため、逆変換では 2 つの結果が発生します。

あいまい変換

単純なテキスト変換と検索。あいまいな変換を実行するには、LSPep2AmbNuc ユーザー定義関数を使用します。

最近似変換

コドン度数表から追加情報が必要です。最近似変換を実行するには、LSPep2ProbNuc ユーザー定義関数を使用します。

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

関連資料:

- 500 ページの『LSPep2AmbNuc ユーザー定義関数』
- 503 ページの『LSPep2AmbNuc ユーザー定義関数のエラー・メッセージ』
- 503 ページの『LSPep2ProbNuc ユーザー定義関数』
- 505 ページの『LSPep2ProbNuc ユーザー定義関数のエラー・メッセージ』
- 501 ページの『LSPep2AmbNuc ユーザー定義関数の例』
- 504 ページの『LSPep2ProbNuc ユーザー定義関数の例』

LSPep2AmbNuc ユーザー定義関数

```
DB2LS.LSPep2AmbNuc(input peptide sequence, filepath to external translation table)
```

input peptide sequence

ペプチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 10890 バイト以下である必要があります。入力データには、標準アミノ酸記号およびあいまいコードを使用します。

filepath to external translation table

カスタマイズした変換表を使用する場合は、その表へのファイル・パス情報を指定してください。パスのストリング値は、255 文字以下にしてください。

スキーマ名は DB2LS です。

LSPep2AmbNuc 関数を使用し、変換表に基づいてペプチド・シーケンスからあいまいなヌクレオチド・シーケンスを作成します。

関数の結果は、データ・タイプ `VARCHAR` の文字ストリングで、実際の長さは 32672 バイト以下になります。この結果は、内蔵または指定した変換表に基づくあいまいなヌクレオチド・シーケンスを示します。

変換表を指定しない場合、関数はデフォルトで表 96 を使用します。

表 96. デフォルトの変換表

アミノ酸記号	省略形	コドン
A	Ala	GCX
B	Asx	RAY
C	Cys	TGY
D	Asp	GAY
E	Glu	GAR
F	Phe	TTY
G	Gly	GGX
H	His	CAY
I	Ile	ATH
K	Lys	AAR
L	Leu	YTX
M	Met	ATG
N	Asn	AAV
P	Pro	CCX
Q	Gln	CAR
R	Arg	MGX
S	Ser	WSX
T	Thr	ACX
V	Val	GTX
W	Trp	TGG
X	Xxx	XXX
Y	Tyr	TAY
Z	Glx	SAR
*	End	TRR

関連資料:

- 503 ページの『LSPep2AmbNuc ユーザー定義関数のエラー・メッセージ』
- 503 ページの『LSPep2ProbNuc ユーザー定義関数』
- 501 ページの『LSPep2AmbNuc ユーザー定義関数の例』

LSPep2AmbNuc ユーザー定義関数の例

この関数は、`values` ステートメントで呼び出せます。入力、以下の例のようにペプチド・シーケンス 1 つです。

```
values db21s.LSPep2AmbNuc('HR');
```

上記の例では、あいまい変換および内蔵の変換表を使用してペプチドをヌクレオチドに変換します。この結果、標準アミノ酸記号から次のようなヌクレオチド・シーケンスが作成されます。

```
CAYMGX
```

以下の例では、あいまい変換と内蔵表を使用してペプチドをヌクレオチドに変換します。

```
values db2ls.LSPep2AmbNuc('SRGFGFITYSHSSMIDEAQKSRPHKIDGRVVEPKRA');
```

この values ステートメントの結果が、次のヌクレオチド・シーケンスです。(シーケンスは、このページに収まるように改行されています。)

```
WSXMGXGGXTTYGGXTTYATHACXTAYWSXCAYWSXWSXATGATHGAYGARGCXCARA  
ARWSXMGXCCXCAYAARATHGAYGGXMGXGTXTXGTXGARCCXAARMGXGCX
```

次の例では、表またはニックネームから抽出された値のセットに適用する関数を示します。

```
SELECT DB2LS.LsPep2AmbNuc(peptide_seq) FROM table protein_table;
```

表 protein_table の peptide_seq 列のデータは、以下のようになります。

表 97. peptide_seq 列のデータ

peptide_seq
GIKEDTEEHHLRDYFE
QKYHTVNGHNCEVRKA
.....

select ステートメントの結果は次のとおりです。

```
GGXATHAARGARGAYACXGARGARCAYCAYYTXMGXGAYTAYTTYGAR  
CARAARTAYCAYACXGTAAAYGGXCAYAAITGYGARGTXMGXAARGCX  
...
```

以下の例では、あいまい変換およびユーザー定義表を使用してペプチドをヌクレオチドに変換します。通常、変換表間の差は、あまりありません。固有な記号が、1つか2つ存在することがあります。これは、より多くのコドンを持つ種や、より少ないコドンを持つ種が存在するためです。例えば、ショウジョウバエはコドン AGG を持ちません。

```
values db2ls.LSPep2AmbNuc('RGNMGGNYGNQNGGGNWNNG',  
                           '%data%transl_table_06.txt')
```

ショウジョウバエ用の変換表を入力した場合、values ステートメントの結果は次のようになります。

```
MGRGGXAAYATGGGXGGXGGXAAYTAYGGXAAYTARAAYGGXGGXGGXAAYTGGAAAYAYGGX
```

関連資料:

- 500 ページの『LSPep2AmbNuc ユーザー定義関数』
- 528 ページの『LSNuc2Pep ユーザー定義関数の例』

LSPep2AmbNuc ユーザー定義関数のエラー・メッセージ

表 98. LSPep2AmbNuc ユーザー定義関数で発行されるメッセージ

エラー・コード	メッセージ	説明
SQL0443N	ルーチン "DB2LS.LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUC") が、診断テキスト "Sequence not valid" とともにエラー SQLSTATE を返しました。SQLSTATE=38608	指定したシーケンスが無効です。
SQL0443N	ルーチン "DB2LS.LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUCUT") が、診断テキスト "No translation found" とともにエラー SQLSTATE を返しました。SQLSTATE=38610	変換表ファイルが空です。
SQL0443N	ルーチン "LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUCUT") が、診断テキスト "Can not open the translation table file" とともにエラー SQLSTATE を返しました。SQLSTATE=38612	指定された変換表ファイルが存在しません。
SQL0443N	ルーチン "DB2LS.LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUCUT") が、診断テキスト "Line too long reading from file" とともにエラー SQLSTATE を返しました。SQLSTATE=38614	ファイルに規定よりも長い行が含まれています。
SQL0443N	ルーチン "DB2LS.LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUCUT") が、診断テキスト "Invalid data file" とともにエラー SQLSTATE を返しました。SQLSTATE=38615	ファイルの形式が無効です。
SQL0443N	ルーチン "LSPEP2AMBNUC" (特定名 "LSPEP2AMBNUCUT") が、診断テキスト "Can't construct the translation table" とともにエラー SQLSTATE を返しました。SQLSTATE=38611	ファイルに無効な記号が検出されました。

関連資料:

- 500 ページの『LSPep2AmbNuc ユーザー定義関数』

LSPep2ProbNuc ユーザー定義関数

►►—DB2LS.LSPep2ProbNuc—(*input peptide sequence*—,filepath to codon frequency table)—►►

input peptide sequence

ペプチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 10890 バイト以下である必要があります。入力データには、標準アミノ酸記号を使用します。

filepath to codon frequency table

コドン度数表です。度数表へのファイル・パス情報を指定してください。パスのストリング値は、255 文字以下にしてください。

スキーマ名は DB2LS です。

LSPep2ProbNuc 関数を使用して、2 番目の引き数で指定されたコドン度数表に基づき、ペプチド・シーケンスから最近似のヌクレオチド・シーケンスを生成します。

関数の結果は、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32672 バイト以下になります。また、コドン度数表を用いた最近似ヌクレオチド・シーケンスを表します。

関連資料:

- 500 ページの『LSPep2AmbNuc ユーザー定義関数』
- 505 ページの『LSPep2ProbNuc ユーザー定義関数のエラー・メッセージ』
- 504 ページの『LSPep2ProbNuc ユーザー定義関数の例』

LSPep2ProbNuc ユーザー定義関数の例

yeast_high.cod 度数表に定義された最近似変換を使用して、ペプチド・シーケンスをヌクレオチド・シーケンスに変換する方法は以下のとおりです。

```
values db21s.LSPep2ProbNuc('RDNNDDN', '%data%yeast_high.cod')
```

この values ステートメントの結果は以下のとおりです。

```
AGAGACAATAACGACGATGATAAC
```

同じステートメントの 2 回目の実行では以下のストリングが形成されます。

```
AGAGATAATAACGACGATGACAAC
```

同じステートメントの 3 回目の実行では以下のランダム値のストリングが形成されます。

```
AGAGATAAACAACGACGACGATAAT
```

太字で強調表示されたコドンは、現在と以前に行われた変換の相違を示します。

単一の values ステートメントの結果は、LSPep2ProbNuc 関数が以前の統計に基づき有力な記号を選択することを表します。この点は、変換可能なものがより多く存在する場合にあいまい記号を使用する LSPep2AmbNuc 関数と異なります。

LSPep2ProbNuc 関数は、各記号の最近似変換を選出し、すべての記号を事前選出したセットからランダム変換を使用して置き換えます。度数表に次のようなデータがあると想定します。

表 99. 度数表データのサンプル

アミノ酸	コドン	度数
Ala	GCG	0.17
Ala	GCA	0.13
Ala	GCT	0.17
Ala	GCC	0.53

ペプチド・シーケンスが、4 つの「A」記号 (Ala) を含むと想定します。この関数は、A を GCC に 2 回、そして GCG と GCT にそれぞれ 1 回ずつ変換します。ただし、変換の順序はランダムになります。照会では、最初の A をセット {GCC、GCC、GCG、GCT} からの各変換に変換します。結果出力の DNA シーケンスに

は、常に 2 つの GCC、1 つの GCG、そして 1 つの GCT が表れます。同一シーケンス上で関数を複数回実行すると、値が置き換えられた DNA シーケンスが戻されることがあります。

関連資料:

- 503 ページの『LSPEP2ProbNuc ユーザー定義関数』
- 505 ページの『LSPEP2ProbNuc ユーザー定義関数のエラー・メッセージ』
- 501 ページの『LSPEP2AmbNuc ユーザー定義関数の例』

LSPEP2ProbNuc ユーザー定義関数のエラー・メッセージ

表 100. LSPEP2ProbNuc ユーザー定義関数で発行されるメッセージ

エラー・コード	メッセージ	説明
SQL0443N	ルーチン "DB2LS.LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "Sequence not valid" とともにエラー SQLSTATE を返しました。SQLSTATE=38608	入力されたシーケンスが無効です。
SQL0443N	ルーチン "DB2LS.LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "No translation found" とともにエラー SQLSTATE を返しました。SQLSTATE=38610	コドン度数表ファイルが空です。
SQL0443N	ルーチン "LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "Can't open the translation table file" とともにエラー SQLSTATE を返しました。SQLSTATE=38612	ファイルが存在しません。
SQL0443N	ルーチン "DB2LS.LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "Line too long reading from file" とともにエラー SQLSTATE を返しました。SQLSTATE=38614	ファイルに規定より長い行が含まれます。
SQL0443N	ルーチン "DB2LS.LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "Invalid data file" とともにエラー SQLSTATE を返しました。SQLSTATE=38615	ファイルの形式が無効です。
SQL0443N	ルーチン "LSPEP2PROBNUC" (特定名 "LSPEP2PROBNUC") が、診断テキスト "Can't construct the translation table" とともにエラー SQLSTATE を返しました。SQLSTATE=38611	ファイルに無効な記号が含まれます。

関連資料:

- 503 ページの『LSPEP2ProbNuc ユーザー定義関数』
- 504 ページの『LSPEP2ProbNuc ユーザー定義関数の例』

定義行構文解析のユーザー定義関数

定義行構文解析のユーザー定義関数 - 概要

定義行構文解析のユーザー定義関数は、定義行エレメントを解析します。たとえば、定義行構文解析のユーザー定義関数は次の処理を行います。

- 定義行から解析したシーケンス ID の他のデータ・ソースとの結合を可能にする。
- 'species = "human"' のような定義行部分の述部を評価する。

定義行構文解析関数は、最も一般的な定義行の形式に対応します。例えば、BLAST ラッパーが戻す定義行エレメント、または FASTA 形式のデータ・ファイルに示される定義行エレメントです。

関連概念:

- 122 ページの『定義行の構文解析』

関連資料:

- 508 ページの『LSDeflineParse ユーザー定義関数の例』
- 506 ページの『LSDeflineParse ユーザー定義関数』

LSDeflineParse ユーザー定義関数

各 LSDeflineParse 関数は、NCBI 標準 FASTA シーケンス ID (NSID) およびその記述のフィールドを解析し、表の列へ出力します。コンパウンド定義である定義行は、各行に単一のコンポーネント定義を含む複数行に出力されます。

DB2LS は、定義行構文解析ユーザー定義関数で使用するスキーマ名です。

定義行構文解析ユーザー定義関数は、次のとおりです。

```
▶▶—DB2LS.LSDeflineParse2—(definition line)————▶▶
▶▶—DB2LS.LSDeflineParse3—(definition line)————▶▶
▶▶—DB2LS.LSDeflineParse2_2—(definition line)————▶▶
▶▶—DB2LS.LSDeflineParse2_3—(definition line)————▶▶
▶▶—DB2LS.LSDeflineParse3_3—(definition line)————▶▶
```

definition line

FASTA 形式で定義行を示す有効な文字列です。文字列表記は、データ・タイプ VARCHAR であり、実際の長さが 1024 バイト以下である必要があります。

LSDeflineParse2 ユーザー定義関数:

LSDeflineParse2 は 2 つのフィールドから成る NSID を持つ定義行を解析します。関数の結果は、4 列の表になります。

表 101. *LSDefineParse2* ユーザー定義関数の結果表の列の説明

列名	説明
ROWID	関数が戻す行数 (整数)。
TAG	NSID タグを表す 3 文字以下の VARCHAR。
IDENTIFIER	NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
DESCRIPTION	1019 文字以下の VARCHAR。

LSDefineParse3 ユーザー定義関数:

3 つのフィールドから成る NSID を持つ定義行を解析します。関数の結果は、5 列の表になります。

表 102. *LSDefineParse3* ユーザー定義関数の結果表の列の説明

列名	説明
ROWID	関数が戻す行数 (整数)。
TAG	NSID タグを表す 3 文字以下の VARCHAR。
ACCESSION	NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
LOCUS	NSID の 3 番目の ID フィールドを表す 20 文字以下の VARCHAR。
DESCRIPTION	1017 文字以下の VARCHAR。

LSDefineParse2_2 ユーザー定義関数:

LSDefineParse2_2 は、2 つのフィールドを持つ NSID が連結されたペアから成るコンパウンド ID の定義行を解析します。関数の結果は、6 列の表になります。

表 103. *LSDefineParse2_2* ユーザー定義関数の結果表の列の説明

列名	説明
ROWID	関数が戻す行数 (整数)。
TAG1	1 番目の ID の NSID タグを表す 3 文字以下の VARCHAR。
IDENTIFIER1	1 番目の NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
TAG2	1 番目の ID の NSID タグを表す 3 文字以下の VARCHAR。
IDENTIFIER2	2 番目の NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
DESCRIPTION	1015 文字以下の VARCHAR。

LSDefineParse2_3 ユーザー定義関数:

LSDefineParse2_3 は、2 つのフィールドから成る NSID が 3 つのフィールドから成る NSID と連結されたコンパウンド ID の定義行を解析します。入力定義行での連結順 (2 つのフィールドから成る NSID と 3 つのフィールドから成る NSID の順) は問題ではありません。関数の結果は、7 列の表になります。

表 104. LSDefineParse2_3 ユーザー定義関数の結果表の列の説明

列名	説明
ROWID	関数が戻す行数 (整数)。
TAG1	2 つのフィールド ID の NSID タグを表す 3 文字以下の VARCHAR。
IDENTIFIER	2 つの フィールドから成る NSID の 2 番目の ID を表す 20 文字以下の VARCHAR。
TAG2	3 つのフィールドから成る ID の NSID タグを表す 3 文字以下の VARCHAR。
ACCESSION	3 つのフィールドから成る NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
LOCUS	3 つのフィールドの NSID の 3 番目の ID フィールドを表す 20 文字以下の VARCHAR。
DESCRIPTION	1013 文字以下の VARCHAR。

LSDefineParse3_3 ユーザー定義関数:

LSDefineParse3_3 は、3 つのフィールドを持つ NSID ペアから成るコンパウンド ID の定義行を解析します。関数の結果は、8 列の表になります。

表 105. LSDefineParse3_3 ユーザー定義関数の結果表の列の説明

列名	説明
ROWID	関数が戻す行数 (整数)。
TAG1	1 番目の ID の NSID タグを表す 3 文字以下の VARCHAR。
ACCESSION1	1 番目の NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
LOCUS1	1 番目の NSID の 3 番目の ID フィールドを表す 20 文字以下の VARCHAR。
TAG2	1 番目の ID の NSID タグを表す 3 文字以下の VARCHAR。
ACCESSION2	2 番目の NSID の 2 番目の ID フィールドを表す 20 文字以下の VARCHAR。
LOCUS2	2 番目の NSID の 3 番目の ID フィールドを表す 20 文字以下の VARCHAR。
DESCRIPTION	1014 文字以下の VARCHAR。

関連資料:

- 508 ページの『LSDefineParse ユーザー定義関数の例』

LSDefineParse ユーザー定義関数の例

このトピックでは、LSDefineParse ユーザー定義関数が定義行を結果表へ解析する方法を、7 つの例を使用して解説します。

以下の照会および結果表の例では、LSDefineParse2 ユーザー定義関数が 2 つのフィールドを持つ NSID の定義行を解析する方法を示します。

```
select *
from table(DB2LS.LSDefineParse2(
    '>gi|12346 hypothetical protein 185 -wheat chloroplast')) as t
```

結果表は次のデータを含みます。

表 106. *LSDefineParse2* ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG	gi
IDENTIFIER	12346
DESCRIPTION	hypothetical protein 185 - wheat chloroplast

以下の照会および結果表の例では、*LSDefineParse3* ユーザー定義関数が 3 つのフィールドを持つ NSID を含む定義行を解析する方法を示します。

```
select *
from table(DB2LS.LSDefineParse3('
    >gb|U37104|APU37104 Aethia pusilla cytochrome b gene')) as t
```

結果表は次のデータを含みます。

表 107. *LSDefineParse3* ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG	gb
ACCESSION	U37104
LOCUS	APU37104
DESCRIPTION	Aethia pusilla cytochrome b gene

以下の照会および結果表の例では、*LSDefineParse2* ユーザー定義関数が 2 つのフィールドを持つ NSID のコンパウンド ID を含む定義行を解析する方法を示します。

```
select *
from table(DB2LS.LSDefineParse2_2(
    '>gb|U37104|gim|73401A Aethia pusilla cytochrome b gene')) as t
```

結果表は次のデータを含みます。

表 108. *LSDefineParse2* ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG1	gb
IDENTIFIER1	U37104
TAG2	gim
IDENTIFIER2	73401A
DESCRIPTION	Aethia pusilla cytochrome b gene

以下の照会例では、3つのフィールドを持つ NSID と 2つのフィールドを持つ NSID の連結からなるコンパウンド ID を持つ定義行を含みます。LSDeflineParse2_3 関数が定義行を解析する方法を次に示します。

```
select *
from table(DB2LS.LSDeflineParse2_3('
>gi|12346|gp|CAA44030.1|CHTAHSRA_4
hypothetical protein 185 - wheat chloroplast')) as t
```

結果表は次のデータを含みます。

表 109. LSDeflineParse2 ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG1	gi
IDENTIFIER	12346
TAG2	gp
ACCESSION	CAA44030.1
LOCUS	CHTAHSRA_4
DESCRIPTION	hypothetical protein 185 - wheat chloroplast

以下の照会例では、2つのフィールドから成る NSID が 3つのフィールドから成る NSID に連結したコンパウンド ID の定義行を含みます。LSDeflineParse2_3 関数が定義行を解析する方法を次に示します。

```
select *
from table(DB2LS.LSDeflineParse2_3('
>gp|CAA44030.1|CHTAHSRA_4|gi|12346
hypothetical protein 185 - wheat chloroplast')) as t
```

結果表は次のデータを含みます。

表 110. LSDeflineParse2 ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG1	gi
IDENTIFIER	12346
TAG2	gp
ACCESSION	CAA44030.1
LOCUS	CHTAHSRA_4
DESCRIPTION	hypothetical protein 185 - wheat chloroplast

以下の照会および結果表の例では、LSDeflineParse3_3 ユーザー定義関数が 3つのフィールドを持つ NSID のペアから成るコンパウンド ID を持つ定義行を解析する方法を示します。

```
select * from table(DB2LS.LSDeflineParse3_3('
>dbj|AAD55586.1|AF055084_1|gp|CAA44030.1|CHTAHSRA_4
hypothetical protein 185 - wheat chloroplast')) as t
```

結果表は次のデータを含みます。

表 111. *LSDefineParse3* ユーザー定義関数の結果データ

列名	データ
ROWID	1
TAG1	dbj
ACCESSION1	AAD55586.1
LOCUS1	AF055084_1
TAG2	gp
ACCESSION2	CAA44030.1
LOCUS2	CHTAHSRA_4
DESCRIPTION	hypothetical protein 185 - wheat chloroplast

コンパウンド定義行の解析には、任意の定義行ユーザー定義関数を使用できます。以下の照会例は、コントロール A で分離された複数定義を持つコンパウンド定義行を含みます。このタイプの定義行は、NCBI の非冗長タンパク質データベース *nr* で検出できます。*LSDefineParse2_3* 関数が定義行を解析する方法を次に示します。

```
select *
from table(DB2LS.LSDefineParse2_3(
  >gi|12346|gp|CAA44030.1|CHTAHSRA_4
  hypothetical protein 185 - wheat chloroplast
  ^Agp|CAA44030.1|CHTAHSRA_4|gi|12346
  hypothetical protein 185 - wheat chloroplast')) as t
```

結果表は次のデータを含みます。

表 112. *LSDefineParse2* ユーザー定義関数の結果データ

列名	データ	データ
ROWID	1	2
TAG1	gi	gi
IDENTIFIER	12346	12346
TAG2	gp	gp
ACCESSION	CAA44030.1	CAA44030.1
LOCUS	CHTAHSRA_4	CHTAHSRA_4
DESCRIPTION	hypothetical protein 185 - wheat chloroplast	hypothetical protein 185 - wheat chloroplast

関連資料:

- 506 ページの『*LSDefineParse* ユーザー定義関数』

一般化されたパターン・マッチングのユーザー定義関数

一般化されたパターン・マッチングのユーザー定義関数は、ヌクレオチド・シーケンスまたはペプチド・シーケンスなどの、所定ストリングの特定箇所を識別します。

LSPatternMatch ユーザー定義関数

▶—DB2LS.LSPatternMatch—(input character sequence, pattern)————▶

input character sequence

文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

pattern

有効な任意の Perl 正規表現で指定されたパターンです。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

LSPatternMatch ユーザー定義関数を使用して、指定パターンの入力されたヌクレオチド・シーケンスまたはペプチド・シーケンスを検索できます。

関数は、シーケンス内で最初に一致するパターンの位置を表す整数を結果として戻します。一致するものがない場合、この関数はゼロを戻します。

PROSITE 構文で記述されたパターンがある場合は、それらを LSPrositePattern ユーザー定義関数で Perl 構文に変換できます。変換した構文は、LSPatternMatch ユーザー定義関数で使用できます。

関連資料:

- 512 ページの『LSPatternMatch ユーザー定義関数の例』
- 514 ページの『LSPrositePattern ユーザー定義関数』

LSPatternMatch ユーザー定義関数の例

次の例では、「coward」、「cowage」、「cowboy」、または「cow1」と一致するストリングの開始位置を検索します。

```
values DB2LS.LSPatternMatch('joe the cowboy is next', 'cow(ard|age|boy|1)')
```

関数は文字単位で検索します。この例では、値 9 を戻します。ストリング「cowboy」は、先頭位置を 1 とすると、9 の位置から開始します。

次の例では、「not」および「non」と一致するストリングの開始位置を検索します。

```
values DB2LS.LSPatternMatch('match not and non but  
no match for no or none', 'no[tn]')
```

関数は文字単位で検索します。この例では、値 7 を戻します。ストリング「not」は、先頭位置を 1 とすると、7 の位置から開始します。

LSPatternMatch は、select ステートメントで SQL LIKE ステートメントより影響力のある PERL 構文を使用して、結果をフィルターにかけるのに役立ちます。次の例では、Blast 出力上の LSPatternMatch を使用して、特定のパターンと一致する遺伝子をフィルターにかけます。

```
SELECT BlastOutput.*  
FROM BlastOutput  
WHERE db21s.LSPatternMatch(HSP_H_Seq, 'F[GSTV]PRL') > 0;
```

PROSITE 構文により精通している場合は、上記の照会で LSPrositePattern も使用できます。照会を以下のように変更します。

```
SELECT BlastOutput.*
FROM BlastOutput
WHERE db21s.LSPatternMatch(HSP_H_Seq,
    db21s.LSPrositePattern('F-[GSTV]-P-R-L.') ) > 0;
```

パターン・マッチング関数は、ヌクレオチド・シーケンスまたはペプチド・シーケンスと同様に、他のテキスト・タイプの検索にも役立ちます。パフォーマンスに配慮する場合は、SQL LIKE ステートメントを使用してください。

次の例では、アライメントのサブジェクトまたはターゲット行にあるタンパク質モチーフに基づいて、BLAST hsp アライメントをフィルターにかける照会を示します。この例は、「Zhang,Z., Schaffer,A.A., Miller,W., Madden,T.L., Lipman,D.J., Koonin,E.V. and Altschul,S.F. (1998) Protein sequence similarity searches using patterns as seeds. *Nucl. Acids Res.*, **26**, 3896-3990.」から採用しました。

以下の照会は、P-loop ATPase ドメイン [GA]xxxxGK[ST] を含むサブジェクト・シーケンスであるアライメントのみを戻します。照会は、シノラブディス・エレガンスの細胞死を調整する CED4 を、NCBI の非冗長タンパク質シーケンス・データベースに対する照会シーケンスに使用します。データベースは、GenBank 項目 X69016 の CDS 機能の変換から、Blast 照会を検索します。

```
SELECT HSP_Q_Seq, HSP_Midline, HSP_H_Seq
FROM BlastP b, GBseq gs, gbfeat gf, gbqual gq
WHERE gs.PRIMARYACCESSION = 'X69016' and
    gs.sequencekey = gf.sequencekey and
    gf.featurejoinkey = gq.featurejoinkey and
    gf.FeatureKey = 'CDS' and
    gq.QualifierName = 'translation' and
    gq.QualifierValue = b.BlastSeq and
    db21s.LSPatternMatch(HSP_H_Seq,
    db21s.LSPrositePattern('[GA]-x(4)-G-K-[ST].') ) > 0;
```

正規の照会シーケンスに該当し、推定上の単一ヌクレオチド多形態 (SNP) を含むゲノム・シーケンスから HSP を検索するには、以下の照会例を使用できます。これは、「Extending traditional query-based integration approaches for functional characterization of post-genomic data. (2001) Barbara A Eckman, Anthony S Kosky, and Leonardo A Laroco Jr. *Bioinformatics* 17(7), 587-601.」から採用しています。

照会は、Blast hsp の中央線上のパターン・マッチングを使用して、 ≥ 20 完全一致、単一の不一致、 ≥ 20 完全一致が連続するパターンを検索します。つまり、20 の「|」文字、シングル・スペース、そして 20 の「|」文字がアライメントの中央線にあります。

またこの例では、ヌクレオチド・シーケンスまたはペプチド・シーケンスではない文字列に対する LSPatternMatch ユーザー定義関数の使用法を示します。

```
SELECT HSP_Info, HSP_Midline, HSP_H_Seq
FROM BlastOutput
WHERE db21s.LSPatternMatch(HSP_Midline, '¥|{20} ¥|{20}') > 0;
```

以前の照会を以下のように再書き込みできます。

```
SELECT HSP_Info, HSP_Midline, HSP_H_Seq, func.Position, func.Match
FROM BlastOutput,
  TABLE(SELECT * AS c FROM TABLE(
    LSMultiMatch(HSP_Midline, '¥|{20} ¥|{20}') )
  AS f) AS func
```

2 番目の照会は、一致する文字列およびそのシーケンスでの位置の他、一致する Blast 行を戻します。

BlastOutput は BlastN ニックネームに関するビューです。

関連資料:

- 514 ページの『LSPrositePattern ユーザー定義関数の例』
- 512 ページの『LSPatternMatch ユーザー定義関数』
- 514 ページの『LSPrositePattern ユーザー定義関数』

LSPrositePattern ユーザー定義関数

▶—DB2LS.LSPrositePattern—(pattern)————▶

pattern

Prosite 構文で指定されたパターン・マッチング構文です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

LSPrositePattern ユーザー定義関数を使用して、PROSITE 構文から PERL 構文に変換できます。変換された構文は、LSPatternMatch、LSMultiMatch、および LSMultiMatch3 のユーザー定義関数で使用できます。

関数の結果は、Perl 構文中で正規表現を表す文字ストリングになります。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

関連資料:

- 514 ページの『LSPrositePattern ユーザー定義関数の例』
- 512 ページの『LSPatternMatch ユーザー定義関数』

LSPrositePattern ユーザー定義関数の例

次の例では、PROSITE 構文から PERL 構文へパターンを変換します。

```
values db2ls.LSPrositePattern('[AC]-x-V-x(4)-{ED}.');
```

この関数は、以下の例のように PROSITE 構文の入力パターンを Perl 構文で相当するパターンに変換します。

```
[AC].V.{4}[^ED]
```

次の例では、PROSITE の別の構文パターンを PERL 構文に変換します。

```
values db2ls.LSPrositePattern('<A-x-[ST](2)-x(0,1)-V.');
```

この関数は、入力パターンに基づいて PROSITE 構文からストリングを変換し、以下を戻します。

```
¥AA.[ST]{2}.{0,1}V
```

次の例では、ID 番号 PS01205 を所有する PROSITE データベース項目に対応するパターンを、パターン・マッチング関数で入力として用いられる PERL パターンに変換します。

```
values db21s.LSPrositePattern('R-P-L-[IV]-x-[NS]-F-G-S-[CA]-T-C-P-x-F.')
```

この照会の結果は次のとおりです。

```
RPL[IV].[NS]FGS[CA]TCP.F
```

次の例では、この関数を照会で使用方法を示します。照会は、指定された PROSITE パターンと一致するシーケンスのみを出力します。

```
SELECT H_Accession, HSP_Info, HSP_H_Seq
FROM BlastOutput
WHERE db21s.LSPatternMatch( HSP_H_Seq,
  db21s.LSPrositePattern('R-P-L-[IV]-x-[NS]-F-G-S-[CA]-T-C-P-x-F.')
```

次の例では、ID PS00261 を持つ PROSITE 項目に対応するパターンを変換します。

```
values db21s.LSPrositePattern('C-[STAGM]-G-[HFYL]-C-x-[ST].')
```

この照会の結果は次のとおりです。

```
C[STAGM]G[HFYL]C.[ST]
```

関連資料:

- 512 ページの『LSPatternMatch ユーザー定義関数の例』
- 514 ページの『LSPrositePattern ユーザー定義関数』

正規表現のサポート

正規表現は、オープン・ソース・ソフトウェアである PCRE のライブラリー・パッケージによりサポートされます。これは Philip Hazel 氏によって記述され、著作権はイギリスのケンブリッジ大学に属します。

ソースは <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/> で参照できます。

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

GeneWise ユーザー定義関数

GeneWise ユーザー定義関数は、プロテイン・シーケンスをゲノム・シーケンスにより調整します。

GeneWise は一般的に使用されているコンポーネントで、プロテイン・シーケンスをゲノム DNA シーケンスにより調整します。ただし、イントロンおよびフレームシフト・エラーが発生する可能性があります。

GeneWise へのリンク

このトピックでは、GeneWise ライブラリーへのリンクの手順を解説します。

手順:

GeneWise ライブラリーにリンクするには、次のようにします。

1. Wise2 パッケージのバージョン 2.1.20c を www.ebi.ac.uk/Wise2 からダウンロードします。
2. 必要に応じて、アーカイブをフォルダーに展開します。
3. pthread サポートでパッケージをコンパイルします。このステップに関する詳細は、Wise2 の資料を参照してください。

HP-UX フェデレーテッド・サーバーでは、ソース・コードをコンパイルする前に、コンパイラ・オプションにオプション `+z` を追加する必要があります。このオプションを追加するには、Wise2 パッケージのルート・ディレクトリーからファイル `makefile` を開き、この例に示されているように行に変更を加えてください。

```
CFLAGS = -c -O -DPTHREAD to: CFLAGS = -c -O -DPTHREAD +z
```

4. **make api** のルート・ディレクトリーでこれを実行します。
5. WISE2_HOME 環境変数を、Wise2 パッケージのルート・ディレクトリーを提示するように設定します。
6. `sqllib/cfg/db2dj.ini` ファイルの WISECONFIGDIR 変数を、`wisecfg` サブディレクトリーを示すように設定します。例えば、Wise2 パッケージを `/usr/wise2.1.20c/` にインストールする場合、`WISECONFIGDIR=/usr/wise2.1.20c/wisecfg/` を `db2dj.ini` ファイルに追加します。
7. `sqllib/bin` ディレクトリー中にある **djxlinkLSGeneWise** スクリプトを実行します。
8. **djxlinkLSGeneWise** スクリプトからの出力を検査します。この出力ファイル `djxlinkLSGeneWise.out` は、`sqllib/function` ディレクトリー中にあります。
9. エラーが報告されない場合は、このライブラリーの作成が正常に行われたことを示します。

関連資料:

- 516 ページの『LSGeneWise ユーザー定義関数』

LSGeneWise ユーザー定義関数

▶—DB2LS.LSGeneWise—(*protein sequence*, *DNA_sequence*)—▶

protein sequence

ペプチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

DNA_sequence

ヌクレオチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

表 113 では、LSGeneWise 関数が戻す 1 行の表を説明します。

表 113. LSGeneWise 関数による出力表の列名、タイプ、および説明

列名	タイプ	説明
PROTEIN_OFFSET	INTEGER	アライメントが検索された入力タンパク質シーケンス内の開始オフセットです。
DNA_OFFSET	INTEGER	アライメントが検索された入力 DNA シーケンス内の開始オフセットです。
PROTEIN	VARCHAR(32672)	位置合わせされたシーケンスを表す入力シーケンスのフラグメントです。
SIMILARITY	VARCHAR(32672)	タンパク質シーケンスと DNA シーケンス間の一致を示します。完全一致は、それに対応する記号文字で記されます。スコアが正で完全一致でないものは「+」で示され、不一致はスペースで示されます。
TRANSLATED_DNA	VARCHAR(32672)	DNA シーケンスが変換されたものです。シーケンスには、ダッシュおよび削除やイントロンのような特殊記号が含まれる可能性があります。
DNA	VARCHAR(32672)	フレーム・シフトおよびイントロンなどの特殊マーカーを持つ DNA シーケンスです。

GeneWise プログラムの出力と LSGeneWise UDF の出力の一致点は以下のとおりです。

- GeneWise プログラムで出力されるタンパク質オフセットおよび DNA オフセットは、PROTEIN_OFFSET および DNA_OFFSET 列と一致します。
- GeneWise で先頭行に出力されるタンパク質シーケンスは、PROTEIN 列と一致します。
- GeneWise で出力される 2 行目は、SIMILARITY 列と一致します。
- GeneWise で出力される 3 行目は、TRANSLATED_DNA 列と一致します。
- GeneWise で出力される 4 行目、5 行目、および 6 行目は、縦方向に読み取ることによって DNA 列に結合されます。

LSGeneWise ユーザー定義関数を使用して、イントロンおよびフレームシフト・エラーを考慮しながら、タンパク質シーケンスをゲノム・シーケンスで調整してください。

LSGeneWise ユーザー定義関数の出力に関する詳細は、<http://www.ebi.ac.uk/Wise2> を参照してください。

関連タスク:

- 516 ページの『GeneWise へのリンク』

関連資料:

- 518 ページの『LSGeneWise ユーザ一定義関数の例』

LSGeneWise ユーザ一定義関数の例

次の例は、LSGeneWise ユーザ一定義関数を使用した照会およびその結果を示します。

```
select protein_offset, dna_offset, protein, similarity, translated_dna, dna
from table( db2ls.LSGeneWise( '
VEPKRAVPRQIDSPNAGATVKKLFGALKDDHDEQSIRDYFQHFQGNIVDINIVIDKETGK
KRGFAFVEFDDYDPVDKVVVLQKQHLNGKQVVKKALPKQNDQQGGGGRRGGPGRAGGNR
GNMGGGNYGNQGGGNWNNGGNNWGNR',
'CACTTAACTGTGAAAGATATTTGTTGGTGGCATTAAAGAAGACTGAAGAACATCACCTAAG
AGATTATTTTGAACAGTATGGAAAAATTGAAGTGAATGAAATCATGACTGACCGAGGCAGTGG
CAAGAAAAGGGGCTTTGCCTTRGTAACCTTTGACGACCATGACTCCGTGGATAAGATTGTCAT
TCAGAAATACCATACTGTGAATGGCCACAACGTGAAGTTAGAAAAGCCCTGTCAAAGCAAGA
GATGGCTAGTGCTTCCAGCCAAAGAGGTCGAAGTGGTTCTGGAAACTTTGGTGGTGGTGG
TGGAGGTGGTTTCGGTGGGAATGACAACCTCGGTCGTGGAGGAACTTCAGTGGTGGTGGTYG
CTTTGGTGGCAGCCGTGGTGGTGGTGGATATGGTGGC' ) ) as f;
```

表 114. 結果表

列	データ
PROTEIN_OFFSET	23
DNA_OFFSET	14
PROTEIN	KLFGALKDDHDEQSIRDYFQHFQGNIVDINIVIDKET GKKRGFVEFDDYDPVDKVVVLQKQHLNGKQVVKKALPKQNDQQGGGGRRGGPGRAGGNRGNMGG GNYGNQGGGNWNNGGN
SIMILARITY	K+FVG +K+D +E +RDYF+ +G I I I+ D+ +GKKRGF+V FDD+D VDK+V+QK H +NG +V+KAL KQ RG G GN+GGG G G N+ GGN
TRANSLATED_DNA	KIFVGGIKEDTEEHHLRDYFEQYQKIEVIEIMTDRGSGK KRGFAxVTFDDHDSVDKIVIQKYHTVNGHNCEVRKAL SKQEMASASSQRGRSGS----- GNFGGRRGGGFGGNDNFGRGGN
DNA	aagatatttgggtggcattaaagaagacactgaagaacatcacctaagat...

関連タスク:

- 516 ページの『GeneWise へのリンク』

関連資料:

- 516 ページの『LSGeneWise ユーザ一定義関数』

Motifs ユーザ一定義関数

Motif ユーザ一定義関数は、ヌクレオチド・シーケンスまたはアミノ酸シーケンスのパターンを一致させます。

LSBarcode ユーザ一定義関数

input string sequence

2 つのシーケンス・フラグメント間の HSP の配置を示す有効な文字ストリングです。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

LSBarCode ユーザー定義関数を使用してシーケンスを入力として使用し、スペースや正符号 (+) を除くすべての文字を垂直バー記号 (|) で置き換えることで、別のシーケンスを生成します。

関数の結果は、バーコード・シーケンスを表す変数文字シーケンスになります。

関連資料:

- 519 ページの『LSBarCode ユーザー定義関数の例』
- 520 ページの『LSMultiMatch ユーザー定義関数』
- 522 ページの『LSMultiMatch3 ユーザー定義関数』

LSBarCode ユーザー定義関数の例

この例では、ストリング・シーケンスからバーコードを作成します。

```
values db2ls.LSBarCode(
  'MDY +G++L GN ++ +PASLTK+MT YVV +A+ + +I D+VTVG+DAWA NP ')
```

この values ステートメントの結果は以下のとおりです。

```
||| +|++| || ++ +|||||+|| ||| +|+ + +| |+||||+|||| ||
```

次の例で、この関数のより実際的な使用方法を解説します。BLAST 検索を実行する研究者が、完全一致の中から 25% 以下のプロリンを含む HSP アライメントのみを戻すと想定します。この例では、関数を使用して BLAST が戻すアライメントの完全一致の中からプロリン (記号 P) のパーセンテージを計算します。この例では、LSMultiMatch3 ユーザー定義関数も呼び出します。照会は、match 関数を使用して完全一致を検出します。これは、Blast が常にアライメントにシーケンス・バー (「|」) を戻すとは限らないため、この照会では LSBarCode 関数とともに使用されます。次のようになります。

```
Query:          MDYTTGQILTAGNEHQQRNPASLTKLMTGYVVDRAIDSHRITPDDIVTVGRDAWAKDNPV
Alignment:     MDY +G++L GN ++ +PASLTK+MT YVV +A+ + +I D+VTVG+DAWA NP
Target:        MDYASGKVLAEAGNADEKLDPASLTKIMTSYVVGQALKADKIKLTDMMVTGKDAWATGNPA
```

出力を、確実にバーの正しいシーケンスで位置合わせするには、LSBarCode 関数を使用します。この関数は、スペースおよび正符号を除くすべての文字を垂直バーで置き換えます。

```
SELECT BlastOutput.* , float( p )/ float( m ) AS percent_prolines
FROM
  BlastOutput b,
  table(SELECT COUNT(*) AS p FROM table(
    db2ls.LSMultiMatch3(
      b.HSP_Q_Seq, 'P',
      db2ls.LSBarCode(b.HSP_Midline), '¥|',
      b.HSP_H_Seq, 'P')
```

```

        ) AS f
    ) AS y,
table(SELECT COUNT(*) AS m FROM table(
    db21s.LSMultiMatch3(
        b.HSP_Q_Seq, '.',
        db21s.LSBarcode(b.HSP_Midline), '¥|',
        b.HSP_H_Seq, '.')
    ) AS f
    ) AS z
WHERE float(p) / float(m) < 0.25;

```

この照会では、BlastOutput は Blast ニックネームに関するビューです。また、この照会では、LSMultiMatch3 関数を使用し、完全一致を位置合わせして戻します。最初の使用で「P」記号の完全一致を戻し、2 回目の使用ですべての完全一致を戻します。結果表の行を表 115 に示します。

表 115. 結果行の例

HSP_Q_SEQ	HSP_H_SEQ	HSP_INFO	PERCENT_PROLINES
NIWDFMQGN...	NIWDFMQGN...	一致 = 80/80 (100%)、 肯定 = 80/80 (100%)、 ギャップ = 0/80 (0%)	+2.500000000000000E-002

上記の照会では、「Extending traditional query-based integration approaches for functional characterization of post-genomic data. (2001) Barbara A Eckman, Anthony S Kosky and Leonardo A Laroco Jr. *Bioinformatics* 17(7), 587-601.」から採用しました。

関連資料:

- 522 ページの『LSMultiMatch3 ユーザー定義関数の例』
- 518 ページの『LSBarcode ユーザー定義関数』

LSMultiMatch ユーザー定義関数

▶—DB2LS.LSMultiMatch—(*input nucleotide or peptide sequence, pattern*)—▶

input nucleotide or peptide sequence

ヌクレオチド・シーケンスまたはペプチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

pattern

Perl 言語で示されるパターン・マッチング構文です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

LSMultiMatch ユーザー定義関数を使用して、入力シーケンスでオーバーラップしない各一致に対し、表を戻すことができます。各表は、開始位置とマッチング・シーケンス・フラグメントから構成されます。

関数の結果は、2 列の表になります。最初の列は、シーケンス内パターンの一致の開始位置を表す整数です。2 列目は、マッチング・シーケンス・フラグメントです。

関連資料:

- 521 ページの『LSMultiMatch ユーザー定義関数の例』
- 518 ページの『LSBarCode ユーザー定義関数』
- 522 ページの『LSMultiMatch3 ユーザー定義関数』

LSMultiMatch ユーザー定義関数の例

次の例では、入力データから取得される重複しないすべての一致に対する位置およびマッチング・フラグメントを検索します。

```
SELECT position, match FROM table
  (LSMultiMatch('match not and non but no match for no or none',
    'no[tn] ')) as f
```

この照会は、一致の結果を表す select ステートメントに基づく表を戻します。

表 116. 複数行を戻す LSMultiMatch の結果

POSITION	MATCH
7	not
15	non

LSMultiMatch は、すべての一致に関して、その位置および一致するストリングを戻します。次の例では、特定モチーフを含むシーケンス項目を、Entrez Nucleotide から検索します。照会は、シーケンス ID および一致したシーケンスを出力します。開始および終了のサブパターン「.{0,9}」は、シーケンスの前と後で最大 9 文字と一致する必要があります。照会はこれらの文字も出力します。

```
select SequenceKey, Position, Match from GBSeq,
  table(db21s.LSMultiMatch(Sequence, '.{0,9}(ATG|CGC)ACGGGC.{0,9}') )
  as fmatch
WHERE entrez.contains(KeywordList,
  'Na/K/2Cl cotransporter AND nkcc1 gene') = 1;
```

この照会の結果は次のとおりです。

表 117. Entrez データの検索

SEQUENCEKEY	POSITION	MATCH
N02B59AE0.04DD4E84	1	TGCTTGGTGATGACGGGCTACCCCAAC
N02B59AE0.04DD4E84	91	GGCCATGTTCGCACGGGCTCCAGAAGG
N02B59AE0.04DC5EF4	1	TGCTTGGTGATGACGGGCTACCCCAAC
N02B59AE0.04DC5EF4	91	GGCCATGTTCGCACGGGCTCCAGAAGG

関連資料:

- 520 ページの『LSMultiMatch ユーザー定義関数』
- 518 ページの『LSBarCode ユーザー定義関数』
- 522 ページの『LSMultiMatch3 ユーザー定義関数』

LSMultiMatch3 ユーザー定義関数

▶—DB2LS.LSMultiMatch3—(input string1, pattern1, input string2, pattern2, input string3, pattern3)—▶

input strings

有効な文字ストリングによる、ヌクレオチド・シーケンスまたはペプチド・シーケンス、または Blast 配置からの HSP_Midline ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

pattern

Perl 言語で示されるパターン・マッチング構文です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

LSMultiMatch3 ユーザー定義関数を使用して 3 つのパターンおよび 3 つのストリングを入力し、3 つすべてのストリングが該当するそれぞれのパターンと一致する位置を戻します。このユーザー定義関数を使用してアライメント上に対するパターン・マッチングを実行できます。

関数の結果は、4 列の表になります。最初の列は、すべてのシーケンスにおけるパターンの一致の開始位置を表す整数です。関数は、先頭位置ですべてのストリングを一緒に固定します。2、3、および 4 列目は、マッチング・シーケンス・フラグメントです。

関連資料:

- 522 ページの『LSMultiMatch3 ユーザー定義関数の例』
- 520 ページの『LSMultiMatch ユーザー定義関数』
- 518 ページの『LSBarCode ユーザー定義関数』

LSMultiMatch3 ユーザー定義関数の例

以下の例では、この関数を使用し、Blast により戻された完全一致における特定のアミノ酸記号のパーセンテージを計算します。この例では、LSBarCode ユーザー定義関数も呼び出します。Blast は常に配置のバーのシーケンス (「|」) を戻すとは限らないため、必要になります。次のようになります。

```
Query:          MDYTTGQILTAGNEHQQRNPASLTKLMTGYVVDRAIDSHRITPDDIVTVGRDAWAKDNPV
Alignment:     MDY +G++L GN ++ +PASLTK+MT YVV +A+ + +I D+VTVG+DAWA NP
Target:        MDYASGKVLAEAGNADEKLDPASLTKIMTSYVVVGQALKADKIKLTDMMVTGKDAWATGNPA
```

出力を、確実に正しいバーのシーケンスで位置合わせするには、LSBarCode 関数を使用してそのシーケンスを変換します。この関数は、スペースおよび「+」を除くすべての文字を垂直バーで置き換えます。

```
SELECT BlastOutput.* , float( p )/ float( m ) AS percent_prolines
FROM
  BlastOutput b,
  table(SELECT COUNT(*) AS p FROM table(
    db21s.LSMultiMatch3(
      b.HSP_Q_Seq, 'P',
      db21s.LSBarCode(b.HSP_Midline), '¥|',
      b.HSP_H_Seq, 'P')
```


結果は以下のとおりです。

表 120. 3 つの入力データを使用した複数一致の結果

POSITION	MATCH_1	MATCH_2	MATCH_3
1	cbcc	a	0
7	ccbbbcc	g	6

関連資料:

- 519 ページの『LSBarCode ユーザー定義関数の例』
- 518 ページの『LSBarCode ユーザー定義関数』
- 522 ページの『LSMultiMatch3 ユーザー定義関数』

反転ユーザー定義関数

反転ユーザー定義関数が、ヌクレオチド・シーケンスまたはアミノ酸シーケンスを反転させます。

LSRevComp ユーザー定義関数

▶▶—DB2LS.LSRevComp—(*input nucleotide sequence*)————▶▶

input nucleotide sequence

ヌクレオチド・シーケンスを示す有効な文字ストリング表記です。シーケンスは、IUPAC あいまいコードを含むことができます。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

スキーマ名は DB2LS です。

関数の結果は、実際の長さが 32672 バイト以下であるデータ・タイプ VARCHAR の文字ストリングであり、ヌクレオチド・シーケンスの逆補数を示します。

関連資料:

- 524 ページの『LSRevComp ユーザー定義関数の例』
- 525 ページの『LSRevNuc ユーザー定義関数』
- 526 ページの『LSRevPep ユーザー定義関数』

LSRevComp ユーザー定義関数の例

LSRevComp 関数は、ヌクレオチド・シーケンスを受け入れる任意の内蔵関数を使用するように SQL ステートメント中で使用できます。例えば、以下のようになります。

```
SELECT DB2LS.LSRevComp(:NucSeq) FROM SYSDDUMMY1;
```

この例では、この関数を使用して、ホスト変数からの入力シーケンスの逆補数を戻します。

無効ストリングおよび無効データ・タイプを使用した場合、以下のようなエラー・メッセージが示されます。

SQL0443N Routine "DB2LS.LSREVCOMP" (specific name "LSREVCOMP") has returned an error SQLSTATE with diagnostic text "Sequence not valid". SQLSTATE=38608

入力されたアルファベットが不適切である場合は、例外が発生します。

LSRevComp ユーザー定義関数は照会で以下のように使用されます。

```
SELECT HSP_H_Seq, db21s.LSRevComp(HSP_H_Seq) as REV_HSP_H_Seq
FROM BlastN
WHERE BlastSeq='ccgctagtattggtcaatcttttgatatccaccgaa'
```

照会の結果は以下のとおりです。

HSP_H_SEQ	REV_HSP_H_SEQ
AGTATTGGTCAATCTTTTGAT	ATCAAAAGATTGACCAATACT
TGGTCAATCTTTTGATA	TATCAAAAGATTGACCA
TTGGCCAATCTTTTGATATCC	GGATATCAAAAGATTGGCCAA
TCAATCTTTTGATATCC	GGATATCAAAAGATTGA
GGATATCAAAAGATTGA	TCAATCTTTTGATATCC

5 record(s) selected.

反転関数をその他のライフ・サイエンス・ユーザー定義関数と併用し、次の例のようにヌクレオチド・シーケンスの逆補数を変換できます。

```
values db21s.LSNuc2Pep(
      db21s.LSRevComp('TTTTTCTTATTGTCTTCCTCATCGTATTTCTTATGTTGCTGATGT'))
```

この照会は以下を戻します。

```
TSAT*EIR*GRQ*EK
```

関連資料:

- 524 ページの『LSRevComp ユーザー定義関数』

LSRevNuc ユーザー定義関数

▶▶—DB2LS.LSRevNuc—(*input nucleotide sequence*)————▶▶

input nucleotide sequence

ヌクレオチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ `VARCHAR` であり、実際の長さが 32672 バイト以下である必要があります。ヌクレオチド・シーケンスは、DNA アルファベットすべて、またはその一部である必要があります。

スキーマ名は `DB2LS` です。

関数の結果は、実際の長さが 32672 バイト以下であるデータ・タイプ `VARCHAR` の文字ストリングであり、ヌクレオチド・シーケンスの逆順を示します。

関連資料:

- 526 ページの『LSRevNuc ユーザー定義関数の例』

- 524 ページの『LSRevComp ユーザー定義関数』
- 526 ページの『LSRevPep ユーザー定義関数』

LSRevNuc ユーザー定義関数の例

LSRevNuc 関数は、ヌクレオチド・シーケンスを受け入れる任意の内蔵関数を使用するように、SQL ステートメント中で使用できます。例えば、以下のようになります。

```
SELECT DB2LS.LSRevNuc(:NucSeq) FROM SYSDDUMMY1;
```

この例では、この関数を使用して、ホスト変数からの入力データを反転します。

無効ストリングおよび無効データ・タイプを使用した場合、以下のようなエラー・メッセージが示されます。

```
SQL0443N Routine "DB2LS.LSREVNUC" (specific name "LSREVNUC") has returned
an error SQLSTATE with diagnostic text "Sequence not valid". SQLSTATE=38608
```

照会での LSRevNuc ユーザー定義関数の使用例は以下のとおりです。

```
SELECT HSP_H_Seq, db21s.LSRevNuc(HSP_H_Seq) as REV_HSP_H_Seq
FROM BlastN
WHERE BlastSeq='gtaatacgtaggggctagcgcgggcaactgaagataaagc'
```

照会が戻す変換されたヌクレオチド・シーケンスは以下のとおりです。

HSP_H_SEQ	REV_HSP_H_SEQ
CGCGGGCAAACCTGAAGATAAAGC	CGAAATAGAAGTCAAACGGGGCGC
GCGCTAGCCCCCTACGTATTAC	CATTATGCATCCCCGATCGCG
GTAATACGTAGGGGGCTAGCG	GCGATCGGGGGATGCATAATG
GTAATACGTAGGGGGCTAGCG	GCGATCGGGGGATGCATAATG
GTAATACGTAGGGGGCTAGCG	GCGATCGGGGGATGCATAATG

5 record(s) selected.

関連資料:

- 525 ページの『LSRevNuc ユーザー定義関数』

LSRevPep ユーザー定義関数

▶—DB2LS.LSRevPep—(*input peptide sequence*)—▶

input peptide sequence

ペプチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。入力シーケンスは、プロテイン・アルファベットの一部分である必要があります。

スキーマ名は DB2LS です。

関数の結果は、実際の長さが 32672 バイト以下であるデータ・タイプ VARCHAR の文字ストリングであり、ペプチド・シーケンスの逆順を示します。

関連資料:

- 527 ページの『LSRevPep ユーザー定義関数の例』
- 524 ページの『LSRevComp ユーザー定義関数』
- 525 ページの『LSRevNuc ユーザー定義関数』

LSRevPep ユーザー定義関数の例

LSRevPep 関数は、ペプチド・シーケンスを受け入れる任意の内蔵関数を使用するように、SQL ステートメント中で使用できます。例えば、以下のようになります。

```
SELECT DB2LS.LSRevPep(:NucSeq) FROM SYSDUMMY1;
```

この例では、この関数を使用して、ホスト変数からの入力データを反転します。

無効ストリングおよび無効データ・タイプを使用した場合、以下のようなエラー・メッセージが示されます。

```
SQL0443N Routine "DB2LS.LSREVPEP" (specific name "LSREVPEP") has returned  
an error SQLSTATE with diagnostic text "Sequence not valid". SQLSTATE=38608
```

照会での LSRevPep ユーザー定義関数の使用例は以下のとおりです。

```
SELECT HSP_H_Seq, db21s.LSRevPep(HSP_H_Seq) as REV_HSP_H_Seq  
FROM BlastP  
WHERE BlastSeq='MLCEIECRALSTAHTRLIHFEPDALTYLEGKNIFTEDH'
```

照会が戻す変換されたペプチド・シーケンスは以下のとおりです。

HSP_H_SEQ	REV_HSP_H_SEQ
MLCEIECRALSTAHTRLIHFEPDALTYL...	HDETFINKGELYTLADRPEFDHILRTHATS...
RVVSTEHTRLVTDAYPEFSISFTATKN	NKTATFISISFEPYADTVLRTHETSVVR
STAHIRVLRDMVPGDEITCFYGSEFF	FFESGYFCTIEDGPVMDRLVRIHATS
AHTRPCPDHEPRGVITYL	LYTIVGRPEHDPCCRTHA

4 record(s) selected.

関連資料:

- 526 ページの『LSRevPep ユーザー定義関数』

変換

変換ユーザー定義関数は、ヌクレオチド・シーケンスをペプチド・シーケンスへと変換します。

LSNuc2Pep ユーザー定義関数

```
▶▶DB2LS.LSNuc2Pep(input nucleotide sequence, filepath to external translation table)▶▶
```

input nucleotide sequence

ヌクレオチド・シーケンスを示す有効な文字ストリング表記です。文字ストリング表記は、データ・タイプ `VARCHAR` であり、実際の長さが 32672 バイト以下である必要があります。

filepath to external translation table

カスタマイズした変換表を使用する場合は、その表へのファイル・パス情報を指定してください。パスのストリング値は、255 文字以下にしてください。

スキーマ名は `DB2LS` です。

関数の結果は、10890 バイト以下であるデータ・タイプ `VARCHAR` の文字ストリングであり、ペプチド・シーケンスを示します。

入力データは、IUB 文字セットを使用するヌクレオチド・シーケンスです。この関数では、ヌクレオチド・シーケンスの先頭文字から最初のコドンが開始されると想定します。最初のコドンが先頭文字から開始されない場合は、入力シーケンスに `SUBSTR` 関数を使用してください。

関数の結果は、標準アミノ酸記号を用いたペプチド・シーケンスになります。

機能:

- 入力シーケンスのスペースを削除する。
- 読み取りフレーム外の無関係なヌクレオチドを無視する。
- `NULL` ヌクレオチド・シーケンスを入力した場合、`NULL` 出力を戻す。

関連資料:

- 528 ページの『LSNuc2Pep ユーザー定義関数の例』
- 529 ページの『LSTransAllFrames ユーザー定義関数』

LSNuc2Pep ユーザー定義関数の例

ヌクレオチド・シーケンス・データをペプチド・シーケンスに変換するとします。この例では、ヌクレオチド・シーケンスの先頭文字から最初のコドンが開始すると想定します。

この関数は、`values` ステートメントで呼び出せます。入力は、以下の例のようにヌクレオチド・シーケンス 1 つです。

```
values db2ls.LSNuc2Pep('TTTTTCTTATTGTCTTCCTCATCGTATTTCTTATGTTGCTGATGT')
```

この結果は、標準アミノ酸記号を用いたペプチド・シーケンスになります。

```
FFLLSSSSYFLCC*C
```

+2 読み取りフレームで変換する場合は、以下を使用してください。

```
values LSNuc2Pep(SUBSTR('TTTTTCTTATTGTCTTCCTCATCGTATTTCTTATGTTGCTGATGT',2))
```

ステートメント中の整数は、コドン検索の開始位置を示します。

この関数を照会の述部として使用する場合は以下のようになります。

```
SELECT *
  FROM proteindata
 WHERE peptideseq=DB2LS.LSNuc2Pep('TTTTTCTTATTGTCTTCCTCATCG
                                   TATTTCTTATGTTGCTGATGT');
```

照会の結果を表 121 に示します。

表 121. *LSNuc2Pep* 関数を述部として使用する場合の結果

ID	PROTEINNAME	PEPTIDSEQ
1	proteinA	FSYCLPHRISYVAD

外部変換表を使用してヌクレオチド・シーケンスをペプチド・シーケンスに変換する場合は以下ようになります。最初のパラメーターはヌクレオチド・シーケンス、2 番目のパラメーターは外部変換表へのパスになります。

```
values db2ls.LSNuc2Pep('TTTTTCTTATTGTCTTCCTCATCGTATTTCTTATGTTGCTGATGT',
                      'C:%translation.txt')
```

このように特定の変換表を使用した場合の結果は、以下のようなストリングになります。

```
FSYCLPHRISYVAD
```

次の例では 2 つのユーザー定義関数を組み合わせ、関数の別の使用方法を示します。

```
values DB2LS.LSNuc2Pep(DB2LS.LSRevCompNuc('TTT..'))
```

上記の例は、以下の照会と同じ結果を戻します。

```
select * from table (DB2LS.LSTransAllFrames ('TTT..')) as t
where t.readframe = -1
```

関連資料:

- 526 ページの『LSRevNuc ユーザー定義関数の例』
- 530 ページの『LSTransAllFrames ユーザー定義関数』
- 527 ページの『LSNuc2Pep ユーザー定義関数』

LSTransAllFrames ユーザー定義関数

```
→DB2LS.LSTransAllFrames—(input nucleotide sequence—┐,filepath to external translation table┘)→
```

input nucleotide sequence

ヌクレオチド・シーケンスを示す有効な文字ストリング表記です。入力シーケンスは、IUPAC あいまいコードを含むことができます。文字ストリング表記は、データ・タイプ VARCHAR であり、実際の長さが 32672 バイト以下である必要があります。

filepath to external translation table

カスタマイズした変換表を使用する場合は、その表へのファイル・パス情報を指定してください。パスのストリング値は、255 文字以下にしてください。

スキーマ名は DB2LS です。

LSTransAllFrames ユーザー定義関数を使用して、入力したヌクレオチド・シーケンスから 1 つのペプチド・シーケンス・セットを形成します。これらのペプチド・シーケンスは、入力された 6 つのフレームごとのヌクレオチド・シーケンスの可能な変換形態を示します。この関数は、入力にエラーが含まれる場合や、読み取りフレームが不明な場合に役立ちます。

関数の結果は、2 列の表になります。1 列目は READFRAME という名前で、変換時に使用されるフレームを表します。この列には、変換の開始位置を示す整数値が入ります。負の整数値は、より線の反対側の変換を示します。2 列目は PEPTIDE と呼ばれ、データ・タイプ VARCHAR の文字ストリングが入ります。これは 10890 バイト以下であり、ペプチド・シーケンスを示します。

機能:

- 入力シーケンスのスペースを削除する。
- 読み取りフレーム外の無関係なヌクレオチドを無視する。
- NULL ヌクレオチド・シーケンスを入力した場合、NULL 出力を戻す。

関連資料:

- 530 ページの『LSTransAllFrames ユーザー定義関数』
- 527 ページの『LSNuc2Pep ユーザー定義関数』

LSTransAllFrames ユーザー定義関数

内蔵された変換表を使用して 6 つの読み取りフレームすべてのヌクレオチド・シーケンスを変換するとします。この方法を例示します。

```
SELECT * FROM table(DB2LS.LSTransAllFrames('TTTTTCTTATTGTCTTCCTCATCG
TATTTCTTATGTTGCTGATGT')) as t;
```

この照会は、以下の例のようにペプチドを表に戻します。

表 122. ヌクレオチド・シーケンスの変換の結果

READFRAME	PEPTIDE
1	FLLSSSSYFLCC*C
2	FSYCLPHRISYVAD
3	FLIVFLIVFLMLLM
-1	TSAT*EIR*GRQ*EK
-2	HQQHKKYDEEDNKK
-3	ISNIRNTMRKTIRK

次の例では、カスタマイズされた変換表を使用して 6 つの読み取りフレームすべてのヌクレオチド・シーケンスを変換します。

```
SELECT * FROM table
(DB2LS.LSTransAllFrames
('TTTTTCTTATTGTCTTCCTCATCGTATTTCTTATGTTGCTGATGT',
'C:¥msvs6¥MyProjects¥alin_udf¥test¥files¥translation.txt')) as t;
```

関数に指定した入力シーケンスと変換表が同じであるため、結果表は前の例と同じになります。

次の例では 2 つのユーザー定義関数を組み合わせ、関数の別の使用方法を示します。

```
values DB2LS.LSNuc2Pep(DB2LS.LSRevCompNuc('TTT..'))
```

上記の例は、以下の照会と同じ結果を戻します。

```
select * from table (DB2LS.LSTransAllFrames ('TTT..')) as t
where t.readframe = -1
```

次の例では、LSTransAllFrames 関数の出力から、特定の読み取りフレームを選択します。

```
SELECT * FROM
  TABLE(db21s.LSTransAllFrames('TTTTCTTATTGTCTTCCTCATCG
                                TATTTCTTATGTTGCTGATGT')) AS t
  WHERE t.readframe=-2
```

この照会の結果は次のとおりです。

表 123. Readframe 関数の使用法

READFRAME	PEPTIDE
-2	HQQHKKYDEEDNKK

関連資料:

- 528 ページの『LSNuc2Pep ユーザー定義関数の例』
- 526 ページの『LSRevNuc ユーザー定義関数の例』
- 529 ページの『LSTransAllFrames ユーザー定義関数』

コドン度数表の形式

コドン度数表は、アミノ酸が特定のコドンに逆変換される頻度を示します。

LSPep2ProbNuc ユーザー定義関数はコドン度数表を使用して、入力されたペプチド・シーケンスからヌクレオチド・シーケンスを決定します。

以下に、コドン度数表ファイルの形式について示します。

- 2 つ並んだピリオドは、表の先頭を示します。ピリオドの前のテキストはすべてコメントです。コメントがない場合でも、2 つのピリオドが必要です。
- 表には次の列が含まれます。
 1. Am-Acid: アミノ酸記号を示す 3 文字のコード。
 2. Codon: アミノ酸記号を示すコドン。
 3. Number: 表のコンパイルの対象となった遺伝子中のコドンの発生回数。
 4. x/1000: 遺伝子の変換数 1000 に対するアミノ酸とコドンのペアの予想発生回数。
 5. Fraction: 同義コドン・ファミリーで該当コドンが発生する割合。

この製品のコドン度数表のサンプルは、サブディレクトリー `sqlib/samples/lifesci/ls_udfs` にあります。

関連資料:

- 503 ページの『LSPep2ProbNuc ユーザー定義関数』

コドン度数表の例

図 46 に、コドン度数表のサンプル形式を示します。

Am-Acid	Codon	Number	x/1000	Fraction	..
Gly	GGG	198.00	18.34	0.23	
Gly	GGA	71.00	6.58	0.08	
Gly	GGT	66.00	6.11	0.08	
Gly	GGC	527.00	48.81	0.61	
Glu	GAG	534.00	49.46	0.88	
Glu	GAA	71.00	6.58	0.12	
Asp	GAT	31.00	2.87	0.06	
Asp	GAC	481.00	44.55	0.94	
Val	GTG	396.00	36.68	0.47	
Val	GTA	22.00	2.04	0.03	
Val	GTT	44.00	4.08	0.05	
Val	GTC	384.00	35.57	0.45	
Ala	GCG	446.00	41.31	0.39	
Ala	GCA	71.00	6.58	0.06	
Ala	GCT	116.00	10.74	0.10	
Ala	GCC	503.00	46.59	0.44	
...	(truncated)				

図 46. コドン度数表の例

関連資料:

- 503 ページの『LSPep2ProbNuc ユーザー定義関数』
- 531 ページの『コドン度数表の形式』

変換表の形式

このトピックでは、LSPep2AmbNuc、LSTransAllFrames、および LSNuc2Pep ライフ・サイエンス・ユーザー定義関数が使用する変換表の形式について説明します。

以下に、コドン度数表ファイルの形式について示します。

- 2 つ並んだピリオドは、表の先頭を示します。ピリオドの前のテキストはすべてコメントです。
- 表の各行には、1 文字のアミノ酸記号、3 文字のアミノ酸名、確定コドン、感嘆符、および、あいまいなコドンが示されます。行中の各文字は空白で区切られます。
- 各コドンおよびアミノ酸記号は、1 ファイルに 1 回のみ表記可能です。
- 終止コドンは、記号「*」に変換されます。
- 小文字で表記されるコドンは、開始コドンを示します。
- その他のコドンは大文字で表記されます。
- 変換する対応アミノ酸記号がないコドンは、「X」に変換されます。

この製品の変換表のサンプルは、サブディレクトリー `sqllib/samples/lifesci/ls_udfs` にあります。

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

関連資料:

- 533 ページの『変換表の例』

変換表の例

図 47 に、変換表の例の形式を示します。

標準変換表					
Symbol	3-letter	Codons	! IUPAC	..	
A	Ala	GCT GCC GCA GCG	! GCX		
B	Asx		! RAY		
C	Cys	TGT TGC	! TGY		
D	Asp	GAT GAC	! GAY		
E	Glu	GAA GAG	! GAR		
F	Phe	TTT TTC	! TTY		
G	Gly	GGT GGC GGA GGG	! GGX		
H	His	CAT CAC	! CAY		
I	Ile	ATT ATC ATA	! ATH		
K	Lys	AAA AAG	! AAR		
L	Leu	TTG TTA CTT CTC CTA CTG	! TTR	CTX YTR	; YTX
M	Met	atg	! ATG		
N	Asn	AAT AAC	! AAY		
P	Pro	CCT CCC CCA CCG	! CCX		
Q	Gln	CAA CAG	! CAR		
R	Arg	CGT CGC CGA CGG AGA AGG	! CGX	AGR MGR	; MGX
S	Ser	TCT TCC TCA TCG AGT AGC	! TCX	AGY	; WSX
T	Thr	ACT ACC ACA ACG	! ACX		
V	Val	GTT GTC GTA GTG	! GTX		
W	Trp	TGG	! TGG		
X	Xxx		! XXX		
Y	Tyr	TAT TAC	! TAY		
Z	Glx		! SAR		
*	End	TAA TAG TGA	! TAR TRA		; TRR

図 47. 変換表の例

関連概念:

- 495 ページの『ライフ・サイエンス・ユーザー定義関数の概要』

関連資料:

- 532 ページの『変換表の形式』

第 26 章 KEGG ユーザー定義関数

この章では、KEGG ユーザー定義関数の概要、それらのフェデレーテッド・システムへの追加方法、そして照会における使用方法について説明します。

KEGG ユーザー定義関数 - 概要

KEGG (Kyoto Encyclopedia of Genes and Genomes) は、ゲノム情報を収めたデータベースのスイートです。KEGG ユーザー定義関数は KEGG データベース内のゲノム情報にアクセスするための DB2[®] Information Integrator で提供される関数のセットです。

DB2 Information Integrator が KEGG Web サービス・インターフェースを通じてアクセスすることができる KEGG スイート内のデータベースは、PATHWAY データベースと SSDB (Sequence Similarity Database) の 2 つだけです。PATHWAY データベースとは、代謝パスウェイ、調節パスウェイ、および分子パスウェイを含む生物過程内の分子間相互作用ネットワークに関するデータを集めたものです。SSDB は、完全なゲノム複合体の中の、タンパク質をコードする遺伝子についてのデータを集めたものです。

KEGG ユーザー定義関数は、KEGG API を使用してこれらのデータベースにアクセスします。

KEGG メソッドの多くは、遺伝子やパスウェイなど、値のリストを戻します。これらのメソッドの中には、入力として値のリストが必要なものもあります。複数のメソッドからの複雑な操作の構成を可能にするために、KEGG ユーザー定義関数のほとんどは、表形式とスカラー形式の両方で存在します。表関数は単一値の表を戻します。スカラー関数は値をスペースで区切られたリストとして戻します。

KEGG ユーザー定義関数は、非リレーショナル・ラッパーのライフ・サイエンス・ユーザー定義関数コンポーネントと共にインストールされます。KEGG ユーザー定義関数のインストール後、それらの関数の登録を行わなければなりません。

ネームスペースの競合を避けるため、KEGG ユーザー定義関数のすべては DB2LS スキーマに登録されます。

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』
- 565 ページの『KEGG ユーザー定義関数を使用不可にする』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

機能カテゴリーごとの KEGG ユーザー定義関数

DB2 Information Integrator には、PATHWAY データベースと SSDB (Sequence Similarity Database) にアクセスするための KEGG ユーザー定義関数が含まれています。

次の表は、PATHWAY データベースからデータを取り出すために使用できるユーザー定義関数の一覧です。

表 124. パスウェイ・ユーザー定義関数

説明	関数タイプ	関数名
パスウェイから化合物	スカラー 表	CompoundsByPathwyS CompoundsByPathwyT
パスウェイから酵素	スカラー 表	EnzymesByPathwyS EnzymesByPathwyT
パスウェイから遺伝子	スカラー 表	GenesByPathwyS GenesByPathwyT
化合物からパスウェイ	スカラー 表	PathwysByCompndsS PathwysByCompndsT
酵素からパスウェイ	スカラー 表	PathwysByEnzymesS PathwysByEnzymesT
遺伝子からパスウェイ	スカラー 表	PathwysByGenesS PathwysByGenesT

次の表は、SSDB データベースからデータを取り出すために使用できるユーザー定義関数の一覧です。

表 125. SSDB ユーザー定義関数

説明	関数タイプ	関数名
遺伝子からすべての関係遺伝子	スカラー 表	AllNbrsByGeneS AllNbrsByGeneT
遺伝子からベスト関係の遺伝子	スカラー 表	BestNbrsByGeneS BestNbrsByGeneT
遺伝子からベスト・ベスト関係の遺伝子	スカラー 表	BstBstNbrsByGeneS BstBstNbrsByGeneT
遺伝子から逆ベスト関係の遺伝子	スカラー 表	RevBestNbrsByGeneS RevBestNbrsByGeneT
遺伝子からホモログ	スカラー 表	BestHmlgsByGenesS BestHmlgsByGenesT
遺伝子からベスト・ベスト・ホモログ	スカラー 表	BstBstHmlgByGenesS BstBstHmlgByGenesT
遺伝子からパラログ	スカラー 表	ParalogsByGeneS ParalogsByGeneT
遺伝子から定義	スカラー	DefinitionsByGeneS
モチーフから遺伝子	表	GenesByMotifsT
遺伝子間の Smith-Waterman スコア	スカラー	ScoreBetweenGenesS

GetGenesByMotifsT 以外の SSDB データベース用の表ユーザー定義関数のすべては、固定された出力列のセットを戻します。 GetGenesByMotifsT 関数は、遺伝子ごとに *keggid* VARCHAR(100) と *definition* VARCHAR(1000) を戻します。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

KEGG ユーザー定義関数の関数引き数

KEGG ユーザー定義関数は共通引き数のセットを使用します。 KEGG ユーザー定義関数の引き数を以下のリストで説明します。

cpdlist

化合物のリスト。このリストは、データ・タイプ VARCHAR であり、実際の長さが 1630 バイト以下である必要があります。リスト内の各項目の形式は次のとおりです。

`cpd:compound`

`compound` は化合物 ID です。リスト内の各項目は、スペース、コンマ、またはセミコロンで区切られていなければなりません。

enzymelist

酵素のリスト。このリストは、データ・タイプ VARCHAR であり、実際の長さが 1630 バイト以下である必要があります。リスト内の各項目は、スペース、コンマ、またはセミコロンで区切られていなければなりません。

keggid

文字ストリングとして表現された、生物種ごとの固有な KEGG ID。 KEGG API では、この ID は `gene_ID` といいます。

この ID は、データ・タイプ VARCHAR であり、実際の長さが 100 バイト以下である必要があります。 *keggid* 引き数に使用する形式は `keggorg:gene_name` です。 `keggorg` は 3 文字の KEGG 生物種コードで、文字ストリングとして表現されます。 `gene_name` は遺伝子の名前です。

KEGG データベース内の各生物種には ID が割り当てられています。生物種のリストは頻繁に変更されます。正しい ID については、<http://www.genome.ad.jp/kegg/kegg2.html#genes> でゲノムの現在のリストをチェックしてください。

keggidlist

KEGG ID のリスト。このリストは、区切り文字で区切られている値の文字ストリングです。このリストは、データ・タイプ VARCHAR であり、実際の長さが 1630 バイト以下である必要があります。リスト内の各項目は、スペース、コンマ、またはセミコロンで区切られていなければなりません。

orglist

KEGG 生物種名のリスト。このリストは、区切り文字で区切られている値の文字ストリングです。このリストは、データ・タイプ VARCHAR であ

り、実際の長さが 1630 バイト以下である必要があります。リスト内の各項目は、スペース、コンマ、またはセミコロンで区切られていなければなりません。

このパラメーターが指定されていないと、生物種リスト全体が検索されません。

midlist

モチーフ ID のリスト。 *midlist* は、区切り文字で区切られている値の文字列です。このリストは、データ・タイプ VARCHAR であり、実際の長さが 1630 バイト以下である必要があります。 *midlist* 内の各モチーフ ID は、スペース、コンマ、またはセミコロンで区切られていなければなりません。

各モチーフ ID の形式は `database:entry` です。 `database` はモチーフ・データベースの ID で、 `entry` はそのデータベース内のモチーフ項目です。たとえば、 `pfam` データベースの `DnaJ` 項目のモチーフ ID は `pf:DnaJ` です。

pathwayid

特定のパスウェイを識別する 3 つの部分からなる名前です。この名前は文字列です。この名前は、データ・タイプ VARCHAR であり、実際の長さが 100 バイト以下である必要があります。 *pathwayid* の形式は次のとおりでなければなりません。

```
path:org map
```

org は KEGG 生物種 ID で、 *map* はパスウェイ・マップ ID です。

threshold

Smith-Waterman スコア用の値です。この値は 100 以上でなければなりません。データ・タイプ INTEGER です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

KEGG ユーザー定義関数の登録

KEGG ユーザー定義関数を使用するには、それらの関数を登録する必要があります。

`enable_KEGGFunctions` コマンドが、すべての KEGG ユーザー定義関数をフェデレーテッド・データベースに登録します。これらの関数は、 `DB2LS` というスキーマ名で登録されます。

前提条件:

非リレーショナル・ラッパーのライフ・サイエンス・ユーザー定義関数コンポーネントは、 `DB2 Information Integrator` を使用してインストールする必要があります。

手順:

KEGG ユーザー定義関数を登録するには、`enable_KEGGFunctions` コマンドを実行します。

- Windows が稼働するフェデレーテッド・サーバーでは、このコマンドは `sqllib\bin` ディレクトリーにあります。
- UNIX が稼働するフェデレーテッド・サーバーでは、このコマンドは `sqllib/bin` ディレクトリーにあります。

構文:

```
enable_KEGGFunctions -n dbName -u userID -p password [-force]
```

dbName

関数の登録先であるフェデレーテッド・データベースの名前。

userID

フェデレーテッド・データベース用の有効なユーザー ID。

password

ユーザー ID の有効なパスワード。

force 関数を除去して再登録するのに使用できるオプションのフラグ。このフラグは、関数が破損したか間違えてドロップしてしまった場合に使用します。

ユーザー定義関数登録の例:

次の例は、`enable_KEGGFunctions` コマンド発行時に戻される出力を示しています。

```
C:> enable_KEGGFunctions -n federateddb -u db2admin -p db2admin
```

```
(0) KEGG Functions were found
-- Create KEGG Functions ...
Create KEGG Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

force フラグを使用して、ユーザー定義関数をドロップして登録する例:

次の例は、`force` フラグ付きで `enable_KEGGFunctions` コマンドを発行し、ユーザー定義関数が既に登録されている場合に戻される出力を示しています。

```
C:> enable_KEGGFunctions -n federateddb -u db2admin -p db2admin -force
```

```
(37) KEGG Functions were found
KEGG functions already exist ...
Reinstall KEGG functions ...
-- Drop KEGG Functions ...
Drop KEGG Functions Successfully.
-- Create KEGG Functions ...
Create KEGG Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 「IBM DB2 Information Integrator インストール・ガイド」の『DB2 Information Integrator システムへのリレーショナル・ラッパー、非リレーショナル・ラッパー、およびユーザー定義関数の追加』
- 565 ページの『KEGG ユーザー定義関数を使用不可にする』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

パス・データベース関数

以下のセクションでは、パス・データベースのユーザー定義関数について説明します。これらのセクションには、それぞれの関数の構文および例が含まれています。

GenesByPathwyS ユーザー定義関数

GenesByPathwyS 関数は、あるパスウェイ上のすべての遺伝子を検索するのに使用します。

GenesByPathwyS 関数は、スペースで区切られた遺伝子のリストを返すスカラー関数です。このリストは、データ・タイプ `VARCHAR` の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は `SELECT` ステートメントで使用できます。

構文:

▶▶ DB2LS.GenesByPathwyS (pathwayid) ◀◀

このユーザー定義関数のスキーマ名は `DB2LS` です。

あるパスウェイ上のすべての遺伝子の検索の例:

あるパスウェイ上のすべての遺伝子を検索するには、この関数に `pathwayid` 引き数を指定します。

たとえば、パスウェイ・マップ `00020` 上のすべての大腸菌遺伝子を検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.GenesByPathwyS
            ('path:eco00020')
            AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』

- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

GenesByPathwyT ユーザー定義関数

GenesByPathwyT 関数は、あるパスウェイ上のすべての遺伝子を検索するのに使用します。

GenesByPathwyT 関数は、パスウェイ上の遺伝子の名前の VARCHAR(100) の列を戻す表関数です。戻される列の名前は `gene` です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.GenesByPathwyT(pathwayid)
```

このユーザー定義関数のスキーマ名は DB2LS です。

あるパスウェイ上のすべての遺伝子の検索の例:

特定のパスウェイ上のすべての遺伝子を検索するには、この関数に `pathwayid` 引き数を指定します。

たとえば、パスウェイ・マップ 00020 上のすべての大腸菌遺伝子を検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.GenesByPathwyT  
    ('path:eco00020')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

CompoundsByPathwyS ユーザー定義関数

CompoundsByPathwyS 関数は、あるパスウェイ上のすべての化合物を検索するのに使用します。

CompoundsByPathwyS 関数は、スペースで区切られた化合物のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.CompoundsByPathwyS—(pathwyid)—▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

あるパスウェイ上のすべての化合物の検索の例:

あるパスウェイ上のすべての化合物を検索するには、この関数に *pathwyid* 引き数を指定します。

たとえば、パスウェイ・マップ 00020 上のすべての化合物を検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.CompoundsByPathwyS
            ('path:00020')
            AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

CompoundsByPathwyT ユーザー定義関数

CompoundsByPathwyT 関数は、あるパスウェイ上のすべての化合物を検索するのに使用します。

CompoundsByPathwyT 関数は、パスウェイ上の化合物の名前の VARCHAR(100) の列を戻す表関数です。戻される列の名前は *compound* です。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.CompoundsByPathwyT—(pathwyid)—▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

あるパスウェイ上のすべての化合物の検索の例:

あるパスウェイ上のすべての化合物を検索するには、この関数に *pathwyid* 引き数を指定します。

たとえば、パスウェイ・マップ 00020 上のすべての化合物を検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.CompoundsByPathwyT
                     ('path:00020')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

EnzymesByPathwyS ユーザー定義関数

EnzymesByPathwyS 関数は、あるパスウェイ上のすべての酵素を検索するのに使用します。

EnzymesByPathwyS 関数は、スペースで区切られた酵素のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.EnzymesByPathwyS—(pathwyid)—▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

あるパスウェイ上のすべての酵素の検索の例:

あるパスウェイ上のすべての酵素を検索するには、この関数に *pathwyid* 引き数を指定します。

たとえば、パスウェイ・マップ 00020 上のすべての酵素を検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.EnzymesByPathwyS
             ('path:00020')
             AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

EnzymesByPathwyT ユーザー定義関数

EnzymesByPathwyT 関数は、あるパスウェイ上のすべての酵素を検索するのに使用します。

EnzymesByPathwyT 関数は、パスウェイ上の酵素の名前の VARCHAR(100) 列を戻す表関数です。戻される列の名前は enzyme です。

この関数は SELECT ステートメントで使用できます。

構文:

▶—DB2LS.EnzymesByPathwyT—(*pathwyid*)—————▶

このユーザー定義関数のスキーマ名は DB2LS です。

あるパスウェイ上のすべての酵素の検索の例:

あるパスウェイ上のすべての酵素を検索するには、この関数に *pathwyid* 引き数を指定します。

たとえば、パスウェイ・マップ 00020 上のすべての化合物を検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.EnzymesByPathwyT
('path:00020')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByGenesS ユーザー定義関数

PathwysByGenesS 関数は、指定した遺伝子を含むすべてのパスウェイを検索するのに使用します。

PathwysByGenesS 関数は、スペースで区切られたパスウェイのリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

▶—DB2LS.PathwysByGenesS—(*keggidlist*)—————▶

このユーザー定義関数のスキーマ名は DB2LS です。

特定の遺伝子を含むすべてのパスウェイの検索の例:

ある遺伝子を含むすべてのパスウェイを検索するには、この関数に *keggidlist* 引き数を指定します。

たとえば、大腸菌遺伝子 b0077 と b0078 を含むすべてのパスウェイを検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.PathwysByGenesS
            ('eco:b0077 eco:0078')
            AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByGenesT ユーザー定義関数

PathwysByGenesT 指定した遺伝子を含むすべてのパスウェイを検索するのに使われます。

PathwysByGenesT 関数は、遺伝子に対応するパスウェイの名前の VARCHAR(100) の列を戻す表関数です。戻される列の名前は pathway です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶—DB2LS.PathwysByGenesT—(keggidlist)—————▶▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

特定の遺伝子を含むすべてのパスウェイの検索の例:

ある遺伝子を含むすべてのパスウェイを検索するには、この関数に keggidlist 引き数を指定します。

たとえば、大腸菌遺伝子 b0077 と b0078 を含むすべてのパスウェイを検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.PathwysByGenesT
                     ('eco:b0077 eco:0078')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByCompndsS ユーザー定義関数

PathwysByCompndsS 関数は、指定した化合物すべてを含むすべてのパスウェイを検索するのに使用します。

PathwysByCompndsS 関数は、スペースで区切られたパスウェイのリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.PathwysByCompndsS(cpdlist)▶▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

指定した化合物すべてを含むパスウェイの検索の例:

リストの化合物のすべてを含むすべてのパスウェイを検索するには、この関数に *cpdlist* 引き数を指定します。

たとえば、化合物 C00033 と C00158 を含むすべてのパスウェイを検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.PathwysByCompndsS
            ('cpd:C00033 cpd:C00158')
            AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByCompndsT ユーザー定義関数

PathwysByCompndsT 関数は、指定した化合物すべてを含むすべてのパスウェイを検索するのに使用します。

PathwysByCompndsT 関数は、パスウェイの名前の VARCHAR(100) の列を戻す表関数です。戻される列の名前は *pathway* です。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.PathwysByCompndsT—(*cpdlist*)——▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

指定した化合物すべてを含むパスウェイの検索の例:

リストの化合物のすべてを含むすべてのパスウェイを検索するには、この関数に *cpdlist* 引き数を指定します。

たとえば、化合物 C00033 と C00158 を含むすべてのパスウェイを検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.PathwysByCompndsT
  ('cpd:C00033 cpd:C00158')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByEnzymesS ユーザー定義関数

PathwysByEnzymesS 関数は、指定した酵素すべてを含むすべてのパスウェイを検索するのに使用します。

PathwysByEnzymesS 関数は、スペースで区切られたパスウェイのリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.PathwysByEnzymesS—(*enzymelist*)——▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

指定した酵素すべてを含むすべてのパスウェイの検索の例:

リストの酵素のすべてを含むすべてのパスウェイを検索するには、この関数に *enzymelist* 引き数を指定します。

たとえば、酵素 1.3.99.1 を含むすべてのパスウェイを検索したいとします。

使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.PathwysByEnzymesS
('ec:1.3.99.1')
AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

PathwysByEnzymesT ユーザー定義関数

PathwysByEnzymesT 関数は、指定した酵素すべてを含むすべてのパスウェイを検索するのに使用します。

PathwysByEnzymesT 関数は、パスウェイの名前の VARCHAR(100) の列を戻す表関数です。戻される列の名前は `pathway` です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶—DB2LS.PathwysByEnzymesT—(enzymelist)—▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

指定した酵素すべてを含むすべてのパスウェイの検索の例:

リストの酵素のすべてを含むすべてのパスウェイを検索するには、この関数に *enzymelist* 引き数を指定します。

たとえば、酵素 1.3.99.1 を含むすべてのパスウェイを検索したいとします。

SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.PathwysByEnzymesT
('ec:1.3.99.1')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

Sequence Similarity データベース関数

以下のトピックでは Sequence Similarity データベース (SSDB) のユーザー定義関数を説明します。これらのセクションには、それぞれの関数の構文および例が含まれています。

SSDB データベース照会 (表関数) から戻される列

SSDB (Sequence Similarity Database) 用のユーザー定義表関数の多くは、固定の出力列のセットを戻します。SSDB 表関数を使用する際に、そうした列のサブセットだけが戻されるように指定することができます。戻される列の名前とデータ・タイプを次の表にリストします。

表 126. SSDB データベースの列

列名	列データ・タイプ	説明
keggid1	VARCHAR (100)	照会に指定された遺伝子の ID。
keggid2	VARCHAR (100)	照会から戻された遺伝子の ID。
swscore	DOUBLE	keggid1 と keggid2 の間の Smith-Waterman スコア。
identity	DOUBLE	keggid1 と keggid2 の間の一致パーセント。
overlap	INTEGER	keggid1 と keggid2 の間のオーバーラップ長。
s1_start	INTEGER	keggid1 の配置の開始位置。
s1_end	INTEGER	keggid1 の配置の終了位置。
s2_start	INTEGER	keggid2 の配置の開始位置。
s2_end	INTEGER	keggid2 の配置の終了位置。
best1	INTEGER	keggid1 から keggid2 へのベスト・ヒットを示すフラグ。値 1 は、keggid1 から keggid2 への間にベスト・ヒットの関係があることを示します。値 0 は、keggid1 から keggid2 への間にベスト・ヒットの関係がないことを示します。
best2	INTEGER	keggid2 から keggid1 へのベスト・ヒットを示すフラグ。値 1 は、keggid2 から keggid1 への間にベスト・ヒットの関係があることを示します。値 0 は、keggid2 から keggid1 への間にベスト・ヒットの関係がないことを示します。
def1	VARCHAR (1000)	keggid1 の定義。
def2	VARCHAR (1000)	keggid2 の定義。
length1	INTEGER	keggid1 のアミノ酸の長さ。
length2	INTEGER	keggid2 のアミノ酸の長さ。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連資料:

- 541 ページの『GenesByPathwayT ユーザー定義関数』
- 551 ページの『AllNbrsByGeneT ユーザー定義関数』

- 553 ページの『BstBstNbrsByGeneT ユーザー定義関数』
- 555 ページの『BestNbrsByGeneT ユーザー定義関数』
- 557 ページの『RevBestNbrsByGeneT ユーザー定義関数』
- 559 ページの『ParalogsByGeneT ユーザー定義関数』
- 560 ページの『BestHmlgsByGenesT ユーザー定義関数』
- 562 ページの『BstBstHmlgByGenesT ユーザー定義関数』

AllNbrsByGeneS ユーザー定義関数

AllNbrsByGeneS 関数は、指定した KEGG ID と相同関係にあるすべての生物種を検索するのに使用します。

すべての生物種を検索する代わりに、検索範囲を狭める生物種のリストを指定することができます。

AllNbrsByGeneS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶ DB2LS.AllNbrsByGeneS(keggid, threshold, orglist)
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子と相同関係にある遺伝子すべてを検索する場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.AllNbrsByGeneS
            ('eco:b0002', 200)
            AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値で相同関係にある遺伝子すべてを検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.AllNbrsByGeneS
            ('eco:b0002', 500, 'ece hin')
            AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

AllNbrsByGeneT ユーザー定義関数

AllNbrsByGeneT 関数は、指定した KEGG ID と相同関係にあるすべての生物種を検索するのに使用します。

すべての生物種を検索する代わりに、検索範囲を狭める生物種のリストを指定することができます。

AllNbrsByGeneT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.AllNbrsByGeneT(keggid, threshold [ ,orglist ])
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子と相同関係にある遺伝子すべてを検索する場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.AllNbrsByGeneT  
    ('eco:b0002', 200)) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値で相同関係にある遺伝子すべてを検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.AllNbrsByGeneT
('eco:b0002', 500, 'ece hin')) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』

BstBstNbrsByGeneS ユーザー定義関数

BstBstNbrsByGeneS 関数は、各生物種の遺伝子とベスト・ベストの関係にある遺伝子を検索するのに使用します。

BstBstNbrsByGeneS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ `VARCHAR` の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は `SELECT` ステートメントで使用できます。

構文:

```
DB2LS.BstBstNbrsByGeneS(keggid, threshold, orglist)
```

このユーザー定義関数のスキーマ名は `DB2LS` です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子とベスト・ベストの関係にある遺伝子を検索する場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BstBstNbrsByGeneS
('eco:b0002', 200)
AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値でベスト・ベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BstBstNbrsByGeneS
            ('eco:b0002', 500, 'ece hin')
            AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BstBstNbrsByGeneT ユーザー定義関数

BstBstNbrsByGeneT 関数は、各生物種の遺伝子とベスト・ベストの関係にある遺伝子を検索するのに使用します。

BstBstNbrsByGeneT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
►► DB2LS.BstBstNbrsByGeneT (keggid, threshold, orglist)
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子のベスト・ベストの関係にある遺伝子を検索する場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BstBstNbrsByGeneT
                      ('eco:b0002', 200)) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値でベスト・ベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BstBstNbrsByGeneT
('eco:b0002', 500, 'ece hin')) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BestNbrsByGeneS ユーザー定義関数

BestNbrsByGeneS 関数は、各生物種の遺伝子のベストの関係にある遺伝子を検索するのに使用します。

BestNbrsByGeneS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶ DB2LS.BestNbrsByGeneS(keggid, threshold [, orglist])
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子とベストの関係にある遺伝子を検索する場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BestNbrsByGeneS
('eco:b0002', 200)
AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値でベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BestNbrsByGeneS  
            ('eco:b0002', 500, 'ece hin')  
            AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BestNbrsByGeneT ユーザー定義関数

BestNbrsByGeneT 関数は、各生物種の遺伝子のベストの関係にある遺伝子を検索するのに使用します。

BestNbrsByGeneT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
► DB2LS.BestNbrsByGeneT(keggid, threshold, [orglist])
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子のベストの関係にある遺伝子を検索する場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BestNbrsByGeneT  
                     ('eco:b0002', 200)) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値でベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BestNbrsByGeneT
    ('eco:b0002', 500, 'ece hin')) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

RevBestNbrsByGeneS ユーザー定義関数

RevBestNbrsByGeneS 関数は、各生物種の遺伝子の逆ベストの関係にある遺伝子を検索するのに使用します。

RevBestNbrsByGeneS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
DB2LS.RevBestNbrsByGeneS(keggid, threshold, [orglist])
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子と逆ベストの関係にある遺伝子を検索する場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.RevBestNbrsByGeneS
    ('eco:b0002', 200)
    AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値で逆ベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.RevBestNbrsByGeneS  
            ('eco:b0002', 500, 'ece hin')  
            AS VARCHAR(1000));
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

RevBestNbrsByGeneT ユーザー定義関数

RevBestNbrsByGeneT 関数は、各生物種の遺伝子の逆ベストの関係にある遺伝子を検索するのに使用します。

RevBestNbrsByGeneT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
►► DB2LS.RevBestNbrsByGeneT(keggid, threshold                      orglist)
```

このユーザー定義関数のスキーマ名は DB2LS です。

全生物種リスト検索の例:

全生物種リストを検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、200 より大きいしきい値で大腸菌遺伝子の逆ベストの関係にある遺伝子を検索する場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.RevBestNbrsByGeneT  
                    ('eco:b0002', 200)) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。

特定の生物種のセットの検索の例:

検索する生物種のリストを指定するには、この関数に *keggid* 引き数、*threshold* 引き数、および *orglist* 引き数を指定しなければなりません。

たとえば、500 より大きいしきい値で逆ベストの関係にある遺伝子を検索し、大腸菌遺伝子は 0157 株だけ、ヘモフィラス・インフルエンザ遺伝子はすべての株を戻すようにする場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.RevBestNbrsByGeneT
    ('eco:b0002', 500, 'ece hin')) AS t;
```

大腸菌遺伝子の KEGG コードは *eco* です。遺伝子名は *b0002* です。大腸菌遺伝子の 0157 株の KEGG 名は *ece* です。ヘモフィラス・インフルエンザ遺伝子の KEGG 名は *hin* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

ParalogsByGeneS ユーザー定義関数

ParalogsByGeneS 関数は、生物種内でパラロガス遺伝子を検索するのに使用します。

ParalogsByGeneS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
►—DB2LS.ParalogsByGeneS—(keggid, threshold)—►
```

このユーザー定義関数のスキーマ名は DB2LS です。

パラロガス遺伝子検索の例:

生物種内でパラロガス遺伝子を検索するには、この関数に *keggid* 引き数と *threshold* 引き数だけを指定します。

たとえば、5000 より大きいしきい値で大腸菌生物種内でパラロガス遺伝子を検索する場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.ParalogsByGeneS
            ('eco:b0002', 5000)
            AS VARCHAR(1000));
```

大腸菌生物種の KEGG データベース名は *eco* です。生物種名は *b0002* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

ParalogsByGeneT ユーザー定義関数

ParalogsByGeneT 関数は、生物種内でパラロガス遺伝子を検索するのに使用します。

ParalogsByGeneT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶—DB2LS.ParalogsByGeneT—(keggid, threshold)—————▶▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

パラロガス遺伝子検索の例:

生物種内でパラロガス遺伝子を検索するには、この関数に *keggid* 引き数と *threshold* 引き数を指定します。

たとえば、5000 より大きいしきい値で大腸菌生物種内でパラロガス遺伝子を検索する場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.ParalogsByGeneT
                      ('eco:b0002', 5000)) AS t;
```

大腸菌生物種の KEGG データベース名は *eco* です。生物種名は *b0002* です。

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BestHmlgsByGenesS ユーザー定義関数

BestHmlgsByGenesS 関数は、指定した遺伝子のリストから生物種のベスト相同関係にある遺伝子を検索するのに使用します。

BestHmlgsByGenesS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.BestHmlgsByGenesS(keggorg—, keggidlist)▶▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

パフォーマンスがもっと良い BstBstHmlgByGenesS 関数の方を使用してください。

ベストの相同関係にある遺伝子を検索する例:

遺伝子のリストから生物種のベストの相同関係にある遺伝子を検索するには、この関数に *keggorg* 引き数と *keggidlist* 引き数を指定します。

たとえば、大腸菌遺伝子 b0002、b0003、b0004、および b0005 のリストの中でヘモフィラス・インフルエンザ生物種のための検索を行う場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BestHmlgsByGenesS
            ('hin', 'eco:b0002 eco:b0003 eco:b0004 eco:b0005')
            VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BestHmlgsByGenesT ユーザー定義関数

BestHmlgsByGenesT 関数は、指定した遺伝子のリストから生物種のベストの相同関係にある遺伝子を検索するのに使用します。

BestHmlgsByGenesT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.BestHmlgsByGenesT—(keggorg, keggidlist)————▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

パフォーマンスがもっと良い BstBstHmlgByGenesT 関数の方を使用してください。

ベストの相同関係にある遺伝子を検索する例:

遺伝子のリストから生物種のベストの相同関係にある遺伝子を検索するには、この関数に *keggorg* 引き数と *keggidlist* 引き数を指定します。

たとえば、大腸菌遺伝子 b0002、b0003、b0004、および b0005 のリストの中でヘモフィラス・インフルエンザ生物種のための検索を行う場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BestHmlgsByGenesT
('hin', 'eco:b0002 eco:b0003 eco:b0004 eco:b0005')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BstBstHmlgByGenesS ユーザー定義関数

BstBstHmlgByGenesS 関数は、遺伝子のリストから生物種のベスト・ベストの相同関係にある遺伝子を検索するのに使用します。

BstBstHmlgByGenesS 関数は、スペースで区切られたターゲット ID のリストを戻すスカラー関数です。このリストは、データ・タイプ VARCHAR の文字ストリングで、実際の長さは 32767 バイト以下になります。

この関数は SELECT ステートメントで使用できます。

構文:

▶▶—DB2LS.BstBstHmlgByGenesS—(keggorg, keggidlist)————▶▶

このユーザー定義関数のスキーマ名は DB2LS です。

ベスト・ベスト関係にある遺伝子だけを検索する例:

遺伝子のリストから生物種のベスト・ベスト関係にある遺伝子だけを検索するには、この関数に *keggorg* 引き数と *keggidlist* 引き数を指定します。

たとえば、大腸菌遺伝子 b0002、b0003、b0004、および b0005 のリストの中でヘモフィラス・インフルエンザ生物種のための検索を行う場合、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.BstBstHmlgByGenesS
('hin', 'eco:b0002 eco:b0003 eco:b0004 eco:b0005')
AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

BstBstHmlgByGenesT ユーザー定義関数

BstBstHmlgByGenesT 関数は、遺伝子のリストから生物種のベスト・ベストの相同関係にある遺伝子を検索するのに使用します。

BstBstHmlgByGenesT 関数は、出力列の固定セットを戻す表関数です。

この関数は SELECT ステートメントで使用できます。

構文:

▶—DB2LS.BstBstHmlgByGenesT—(keggorg, keggidlist)—————▶

このユーザー定義関数のスキーマ名は DB2LS です。

ベスト・ベスト関係にある遺伝子だけを検索する例:

遺伝子のリストから生物種のベスト・ベスト関係にある遺伝子だけを検索するには、この関数に *keggorg* 引き数と *keggidlist* 引き数を指定します。

たとえば、大腸菌遺伝子 b0002、b0003、b0004、および b0005 のリストの中でヘモフィラス・インフルエンザ生物種のための検索を行う場合、SELECT ステートメントは次のとおりです。

```
SELECT * FROM TABLE(DB2LS.BstBstHmlgByGenesT
('hin', 'eco:b0002 eco:b0003 eco:b0004 eco:b0005')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』
- 549 ページの『SSDB データベース照会 (表関数) から戻される列』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』

- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

ScoreBetweenGenesS ユーザー定義関数

ScoreBetweenGenesS 関数は、2 つの遺伝子間の Smith-Waterman スコアを判断するのに使用します。

ScoreBetweenGenesS 関数は、データ・タイプ DOUBLE の単一値を戻すスカラー関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.ScoreBetweenGenesS(keggid1, keggid2)
```

このユーザー定義関数のスキーマ名は DB2LS です。

2 つの遺伝子間の Smith-Waterman スコアの判断の例:

2 つの遺伝子間の Smith-Waterman スコアを判断するには、各遺伝子の *keggid* を指定します。

たとえば、大腸菌遺伝子 b0002 と b3940 の間の Smith-Waterman スコアを判断するには、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.ScoreBetweenGenesS
           ('eco:b0002', 'eco:b3940')
           AS DOUBLE);
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

DefinitionsByGeneS ユーザー定義関数

DefinitionsByGeneS 関数は、遺伝子の定義を検索するのに使用します。

DefinitionsByGeneS 関数は、データ・タイプ VARCHAR(1000) の単一値を戻すスカラー関数です。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.DefinitionsByGeneS(keggid)
```

このユーザー定義関数のスキーマ名は DB2LS です。

遺伝子の定義の検索の例:

遺伝子の定義を検索するには、*keggid* を指定します。

たとえば、大腸菌遺伝子 *b0002* の定義を戻すには、使用する文節は次のとおりです。

```
VALUES CAST(DB2LS.DefinitionsByGeneS
            ('eco:b0002')) AS t;
        AS VARCHAR(1000));
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

GenesByMotifsT ユーザー定義関数

GenesByMotifsT 関数は、指定したモチーフのリストのすべてのモチーフを含む遺伝子を検索します。

GenesByMotifsT 関数は、表を戻す表関数です。この表には、遺伝子ごとに *keggid* VARCHAR(100) と *definition* VARCHAR(1000) が含まれます。

この関数は SELECT ステートメントで使用できます。

構文:

```
▶▶ DB2LS.GenesByMotifsT(midlist)▶▶
```

このユーザー定義関数のスキーマ名は DB2LS です。

リストのすべてのモチーフを含む遺伝子の検索の例:

リストのすべてのモチーフを含む遺伝子を検索するには、*midlist* 引き数を指定します。

たとえば、*Pfam 'DnaJ'* と *Prosite 'DNAJ_2'* の両方のモチーフを含む遺伝子を見つけるには、SELECT ステートメントは次のようになります。

```
SELECT * FROM TABLE(DB2LS.GenesByMotifsT
                      ('pf:DnaJ ps:DNAJ_2')) AS t;
```

関連概念:

- 535 ページの『KEGG ユーザー定義関数 - 概要』

関連タスク:

- 538 ページの『KEGG ユーザー定義関数の登録』

関連資料:

- 537 ページの『KEGG ユーザー定義関数の関数引き数』
- 536 ページの『機能カテゴリーごとの KEGG ユーザー定義関数』

KEGG ユーザー定義関数を使用不可にする

KEGG ユーザー定義関数をもう使用しない場合、一時的に関数を使用不可にすることも、フェデレーテッド・データベースから恒久的に削除してしまうことも可能です。

KEGG ユーザー定義関数を使用不可にする場合、関数をフェデレーテッド・データベースに登録することにより、再び使用可能にすることができます。

関数をフェデレーテッド・データベースから恒久的に削除する場合は、関数をアンインストールする必要があります。

手順:

KEGG ユーザー定義関数を使用不可にするには、`disable_KEGGFunctions` コマンドを実行します。

- Windows が稼働するフェデレーテッド・サーバーでは、このコマンドは `sqllib\bin` ディレクトリーにあります。
- UNIX が稼働するフェデレーテッド・サーバーでは、このコマンドは `sqllib/bin` ディレクトリーにあります。

構文:

```
disable_KEGGFunctions -n dbName -u userID -p password
```

dbName

関数を使用不可にするフェデレーテッド・データベースの名前。

userID

フェデレーテッド・データベース用の有効なユーザー ID。

password

ユーザー ID の有効なパスワード。

KEGG ユーザー定義関数を使用不可にする例:

次の例は、`disable_KEGGFunctions` コマンド発行時に戻される出力を示しています。

```
C:>disable_KEGGFunctions -n federateddb -u db2admin -p db2admin
```

```
(37) KEGG Functions were found
    -- Drop KEGG Functions ...
    Drop KEGG Functions Successfully.
```

```
*** Please allow a few seconds to clean up the system .....
```

関連タスク:

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『リレーショナル・ラッパー、非リレーショナル・ラッパー、およびライフ・サイエンス・ユーザー定義関数の除去 (Windows)』

- 「*IBM DB2 Information Integrator* インストール・ガイド」の『ラッパー、ユーザー定義関数、およびラッパー開発キットの除去 (UNIX)』
- 538 ページの『KEGG ユーザー定義関数の登録』

第 5 部 参照情報

第 27 章 非リレーショナル・データ・ソースでサポートされるデータ・タイプ

次のセクションには、非リレーショナル・ラッパーによってサポートされるデータ・タイプをリストしています。

非リレーショナル・データ・ソースでサポートされるデータ・タイプ

ほとんどの非リレーショナル・データ・ソースに関しては、データ・ソースにアクセスするためのニックネームを作成する際に、データ・タイプを含めた列情報を指定する必要があります。

ある種の非リレーショナル・ラッパーは、データ・ソースにアクセスするために必要なすべての列を作成します。これらは、**固定列** と呼ばれます。他のラッパーでは、**CREATE NICKNAME** ステートメントで列のデータ・タイプの一部またはすべてを指定することができます。

以下のセクションでは、データ・タイプを指定できるラッパーと、ラッパーがサポートするデータ・タイプをリストしています。

BioRS ラッパーによってサポートされるデータ・タイプ

次の表には、BioRS ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 127. DB2 データ・タイプにマップされる BioRS データ・タイプ

BioRS データ・タイプ	DB2 データ・タイプ
AUTHOR	CHARACTER、CLOB、VARCHAR
DATE	CHARACTER、CLOB、VARCHAR
NUMBER	CHARACTER、CLOB、VARCHAR
REFERENCE	CHARACTER、CLOB、VARCHAR
TEXT	CHARACTER、CLOB、VARCHAR

CLOB データ・タイプで許容される最大長は 5 MB です。

BLAST ラッパーによってサポートされるデータ・タイプ

BLAST ラッパーが作成する固定列には、ある種のデータ・タイプが自動的に設定されます。

定義行フィールドには、ニックネームの作成時に割り当てることができます。定義行の列の中のデータがローカル列のデータ・タイプと互換性がない場合は、エラーが発生する可能性があります。たとえば、INTEGER タイプの定義行の列を定義しているとき、その列の中に数値以外の値があるなら、エラーが戻されます。

次の表には、BLAST ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 128. DB2 データ・タイプにマップされる BLAST データ・タイプ

BLAST データ・タイプ	DB2 データ・タイプ
定義行	CLOB
	CLOB データ・タイプで許容される最大長は 5 MB です。
定義行	DOUBLE
定義行	FLOAT
定義行	INTEGER
定義行	VARCHAR

Documentum ラッパーによってサポートされるデータ・タイプ

次の表には、Documentum ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 129. DB2 データ・タイプにマップされる Documentum データ・タイプ

Documentum データ・タイプ	DB2 データ・タイプ
DOUBLE	DOUBLE、FLOAT、INTEGER、SMALLINT
ID	CHARACTER (16)
INTEGER	DOUBLE、FLOAT、INTEGER、SMALLINT
STRING (255 文字まで)	CHAR、VARCHAR
TIME	CHAR、DATE、TIMESTAMP、VARCHAR

Entrez ラッパーによってサポートされるデータ・タイプ

次の表には、Entrez ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 130. DB2 データ・タイプにマップされる Entrez データ・タイプ

Entrez データ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CLOB
	CLOB データ・タイプで許容される最大長は 5 MB です。
日付	DATE
数値	DECIMAL
数値	DOUBLE
整数	INTEGER
数値	REAL
整数	SMALLINT
時刻	TIMESTAMP
文字	VARCHAR

Excel ラッパーによってサポートされるデータ・タイプ

次の表には、Excel ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 131. DB2 データ・タイプにマップされる Excel データ・タイプ

Excel データ・タイプ	DB2 データ・タイプ
日付	DATE
数値	DOUBLE
数値	FLOAT (n)。 n は ≥ 25 、 ≤ 53 。
整数	INTEGER
文字	VARCHAR

Extended Search ラッパーによってサポートされるデータ・タイプ

次の表には、Extended Search ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 132. DB2 データ・タイプにマップされる Extended Search データ・タイプ

Extended Search データ・タイプ	DB2 データ・タイプ
Date	DATE
Double	DOUBLE
Integer	INTEGER
String	VARCHAR

HMMER ラッパーによってサポートされるデータ・タイプ

次の表には、HMMER ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 133. DB2 データ・タイプにマップされる HMMER データ・タイプ

HMMER データ・タイプ	DB2 データ・タイプ
文字	CLOB
	CLOB データ・タイプで許容される最大長は 5 MB です。
文字	DOUBLE
文字	FLOAT
文字	INTEGER
文字	VARCHAR

表構造ファイルのラッパーによってサポートされるデータ・タイプ

次の表には、表構造ファイルのラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 134. DB2 データ・タイプにマップされる表構造ファイルのデータ・タイプ

表構造ファイルのデータ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CLOB
	CLOB データ・タイプで許容される最大長は 5 MB です。
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR

Web サービス・ラッパーによってサポートされるデータ・タイプ

次の表には、Web サービス・ラッパーによってサポートされる DB2 データ・タイプをリストしています。Web サービス・ラッパーは、XML データ・タイプを使用します。

表 135. DB2 データ・タイプにマップされる、Web サービス・ラッパー用の XML データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CHARACTER FOR BIT DATA
文字	CLOB
日付	DATE
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA

WebSphere Business Integration ラッパーによってサポートされるデータ・タイプ

次の表には、WebSphere Business Integration ラッパーによってサポートされる DB2 データ・タイプをリストしています。WebSphere Business Integration ラッパーは、XML データ・タイプを使用します。

表 136. DB2 データ・タイプにマップされる、WebSphere Business Integration ラッパー用の XML データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CHARACTER FOR BIT DATA
文字	CLOB
日付	DATE
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA

XML ラッパーによってサポートされるデータ・タイプ

次の表には、XML ラッパーによってサポートされる DB2 データ・タイプをリストしています。

表 137. DB2 データ・タイプにマップされる、XML ラッパー用の XML データ・タイプ

XML データ・タイプ	DB2 データ・タイプ
文字	CHARACTER
文字	CHARACTER FOR BIT DATA
文字	CLOB
	CLOB データ・タイプで許容される最大長は 5 MB です。
日付	DATE
数値	DECIMAL
数値	DOUBLE
数値	FLOAT
整数	INTEGER
数値	REAL
整数	SMALLINT
文字	VARCHAR
文字	VARCHAR FOR BIT DATA

関連概念:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・サーバーでのデータ・タイプ・マッピング』
- 「フェデレーテッド・システム・ガイド」の『データ・タイプ・マッピングとフェデレーテッド・データベース・グローバル・カタログ』

- ・ 「フェデレーテッド・システム・ガイド」の『非リレーショナル・データ・ソースのデータ・タイプ・マッピング』

第 28 章 ニックネームの変更

この章では、既に登録済みのニックネームを変更する方法を説明します。

ニックネームの変更

ニックネームとは、データ・ソースにあるアクセス対象のオブジェクトを参照するために使用する ID です。

以下の作業を行うときに、ニックネームを変更できます。

- データ・ソース・オブジェクトの列のローカル列名を変更する
- データ・ソース・オブジェクトの列のローカル・データ・タイプを変更する
- ニックネームと列のオプションを追加、設定、またはドロップする
- 主キーを追加またはドロップする
- 1 つ以上の固有な制約、参照制約、またはチェック制約を追加またはドロップする
- 1 つ以上の参照制約、チェック制約、または機能従属関係制約を変更する

前提条件:

ステートメントの許可 ID によって保持されている特権には、少なくとも以下のいずれかが含まれていなければなりません。

- SYSADM または DBADM 権限
- ステートメントで指定されたニックネームに対する ALTER 特権
- ステートメントで指定されたニックネームに対する CONTROL 特権
- ニックネームにスキーマ名がある場合は、スキーマに対する ALTERIN 特権
- ニックネームのカタログ・ビューの DEFINER 列に記録された、ニックネームの定義者

制約事項:

『ニックネームの変更に関する制限』のトピックを参照してください。

手順:

ニックネームは、DB2 コントロール・センターまたは DB2 コマンド行から変更できます。

DB2 コントロール・センターからこのタスクを行うには、以下のようになります。

1. 「ニックネーム」フォルダーを選択します。
2. 変更するニックネームを右クリックして、「**変更**」をクリックします。「ニックネームの変更」ノートブックがオープンします。
3. 「ニックネーム」ページで、グローバル・カタログに保管されている列のローカル列名、ローカル・データ・タイプ、または列オプションを変更します。
4. 「キー」ページで、ニックネームの参照保全制約を設定します。主キー、ユニーク・キー、または外部キー制約を設定できます。

5. 「チェック制約」ページでは、ニックネームのチェック制約、または機能従属関係制約を設定します。
6. 「設定」ページで、ニックネームのニックネーム・オプションを設定します。
7. **「OK」**をクリックして、ニックネームを変更し、ノートブックをクローズします。

ニックネーム・オプションの中には必須のものもあり、それらはドロップできません。すでに特定のニックネーム・オプションが設定されている場合には、それ以外のニックネーム・オプションを追加することはできません。各オプションに関する説明のリストについては、『フェデレーテッド・システムのニックネーム・オプション』および『フェデレーテッド・システムのニックネーム列オプション』を参照してください。

この作業を DB2 コマンド行から行うには、ALTER NICKNAME ステートメントに適切なパラメーター・セットを指定して発行します。

データ・ソースのオブジェクト構造や内容が大きく変更された場合は、ニックネーム統計を更新してください。複数の行が追加または削除されたときには、大きな変更とみなします。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『ニックネームの情報制約』
- 「フェデレーテッド・システム・ガイド」の『ニックネーム統計の更新機能 - 概要』

関連タスク:

- 580 ページの『ニックネーム・オプションの変更』
- 583 ページの『データ・ソース・オブジェクトのローカル・タイプの変更』
- 578 ページの『ニックネームの列名の変更』
- 581 ページの『ニックネーム列オプションの変更』

関連資料:

- 576 ページの『ニックネームの変更に関する制限』
- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』
- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』
- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』

ニックネームの変更に関する制限

ニックネームを変更する場合、以下の制限が適用されます。

列名 ALTER NICKNAME ステートメントを使用して、以下のデータ・ソースの列名を変更することはできません。ニックネームをドロップし、正しい列名で再びニックネームを作成しなければなりません。

- BLAST
- Documentum

- HMMER

列オプション

以下のオプションのいずれかが列に設定されている場合、その列に他のオプションを追加することはできません。

- SOAPACTIONCOLUMN
- URLCOLUMN
- PRIMARY_KEY
- FOREIGN_KEY

BioRS の場合

- ELEMENT_NAME オプションを使用して、列の元素名を変更する場合、新しい名前が正しいかどうかを確認するための検査は行われません。オプションが誤っていると、照会で列が参照された時にエラーになる可能性があります。
- IS_INDEXED 列オプションへの変更を行う場合、BioRS サーバーで変更が検査されることはありません。オプションが誤っていると、照会で列が参照された時にエラーになる可能性があります。

データ・タイプ

- 列のデータ・タイプを変更する場合、新しいデータ・タイプは、対応するデータ・ソース列または元素のデータ・タイプと互換性がなければなりません。ローカル・データ・タイプを、リモート・データ・タイプと非互換のデータ・タイプに変更すると、予測不能なエラーが生じる場合があります。
- *local_data_type* は、LONG VARCHAR、LONG VARGRAPHIC、DATALINK またはユーザー定義のデータ・タイプにすることはできません。
- *data_source_data_type* は、ユーザー定義のタイプにすることはできません。
- 非リレーショナル・データ・ソースの中には、既存のローカル・タイプのオーバーライドや新しいローカル・タイプの作成を行えないものがあります。この制限については、特定のデータ・ソース・ラッパーの資料を調べてください。
- 列のデータ・タイプのローカル指定が変更された場合、フェデレーテッド・データベース・マネージャーは、その列について収集された統計 (HIGH2KEY や LOW2KEY など) をすべて無効にします。
- 特定のデータ・ソース・オブジェクトがニックネームを使用してアクセスされた場合、ローカル・タイプがそのオブジェクトに設定されます。同じデータ・ソース・オブジェクトに、デフォルトのデータ・タイプ・マッピングを使用する別のニックネームを付けることができます。

索引 ALTER NICKNAME ステートメントは、フェデレーテッド・データベース内に新しいデータ・ソース索引を登録するためには使用できません。「索引の指定」を作成するには、CREATE INDEX ステートメントに SPECIFICATION ONLY 文節を指定します。

LOCAL NAME および LOCAL TYPE パラメーター

- 以下の場合、ALTER NICKNAME ステートメントを使用して、ニックネーム内の列のローカル名またはデータ・タイプを変更することはできません。
 - ニックネームがビュー、SQL メソッド、または SQL 関数で使用されている。
 - ニックネームに対して情報制約を定義している。
- ALTER NICKNAME ステートメントで、LOCAL NAME パラメーター、LOCAL TYPE パラメーター、またはこの両方を指定する必要もある場合は、最後に federated_column_options 文節を指定する必要があります。

ニックネーム

ALTER NICKNAME ステートメントを使用して、BioRS ニックネームによって参照される、または BioRS ニックネームで使用される BioRS データ・バンクの名前を変更することはできません。BioRS データ・バンクの名前が変更される場合、ニックネームをドロップして、再びそのニックネームを作成しなければなりません。

作業単位

フェデレーテッド・サーバーは、以下の条件のいずれかでは、指定の作業単位内で ALTER NICKNAME ステートメントを処理できません。

- ALTER NICKNAME ステートメントで参照されているニックネームが、同じ作業単位内にオープン・カーソルを持っている場合
- ALTER NICKNAME ステートメントで参照されているニックネームに関して、同じ作業単位で挿入、削除、または更新が発行される場合
- 非リレーショナル・データ・ソースで、ALTER NICKNAME ステートメントが、同じ作業単位内で SELECT ステートメントによって参照されているニックネームを参照する場合

関連タスク:

- 580 ページの『ニックネーム・オプションの変更』
- 583 ページの『データ・ソース・オブジェクトのローカル・タイプの変更』
- 575 ページの『ニックネームの変更』
- 578 ページの『ニックネームの列名の変更』
- 581 ページの『ニックネーム列オプションの変更』

ニックネームの列名の変更

ニックネームを作成すると、データ・ソース・オブジェクトに関連付けられている列名は、フェデレーテッド・データベースに保管されます。データ・ソースには、ラッパーが列名を指定するものと、ニックネーム作成時にユーザーが列名を指定しなければならないものがあります。

ニックネームを変更すると、列名を変更できます。

前提条件:

ステートメントを発行する 許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- SYSADM または DBADM 権限
- ステートメントで指定されたニックネームに対する ALTER 特権
- ステートメントで指定されたニックネームに対する CONTROL 特権
- ニックネームにスキーマ名がある場合は、スキーマに対する ALTERIN 特権
- ニックネームのカタログ・ビューの DEFINER 列に記録された、ニックネームの定義者

制約事項:

『ニックネームの変更に関する制限』のトピックを参照してください。

手順:

列名の変更は、DB2 コントロール・センターまたは DB2 コマンド行プロセッサで行えます。

DB2 コントロール・センターからこのタスクを行うには、以下のようになります。

1. 「ニックネーム」フォルダーを選択します。
2. 変更するニックネームを右クリックして、「変更」をクリックします。「ニックネームの変更」ノートブックがオープンします。
3. 「ニックネーム」ページで、変更する列を選択して「変更」をクリックします。「列の変更」ウィンドウがオープンします。
4. 列名を入力します。
5. 「OK」をクリックして列名を変更し、ウィンドウをクローズします。
6. 「OK」をクリックして、ニックネームを変更し、ノートブックをクローズします。

この作業を DB2 コマンド行から行うには、ALTER NICKNAME ステートメントを発行します。

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

例: ニックネーム列のローカル名の変更:

たとえば、DB2 UDB for z/OS 表のニックネーム Z_EMPLOYEES に EMPNO という名前の列が含まれているとします。このとき、ニックネームを変更して、ユーザーが作業するローカル列名を、EMPNO ではなく Employee_Number にするには、次のステートメントを発行します。

```
ALTER NICKNAME Z_EMPLOYEES ALTER COLUMN EMPNO
LOCAL NAME "Employee_Number"
```

関連タスク:

- 575 ページの『ニックネームの変更』

関連資料:

- 576 ページの『ニックネームの変更に関する制限』
- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』

ニックネーム・オプションの変更

ニックネーム・オプションとは、`CREATE NICKNAME` および `ALTER NICKNAME` ステートメントを発行する時にニックネームに指定する、パラメータのことです。

`ALTER NICKNAME` ステートメントを使用することにより、ニックネーム・オプションを追加、設定、またはドロップできます。

前提条件:

ステートメントを発行する許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- `SYSADM` または `DBADM` 権限
- ステートメントで指定されたニックネームに対する `ALTER` 特権
- ステートメントで指定されたニックネームに対する `CONTROL` 特権
- ニックネームにスキーマ名がある場合は、スキーマに対する `ALTERIN` 特権
- ニックネームのカatalog・ビューの `DEFINER` 列に記録された、ニックネームの定義者

制約事項:

『ニックネームの変更に関する制限』のトピックを参照してください。

手順:

列名の変更は、`DB2 コントロール・センター` または `DB2 コマンド行プロセッサ` で行えます。

`DB2 コントロール・センター` からこのタスクを行うには、以下のようになります。

1. 「ニックネーム」フォルダーを選択します。
2. 変更するニックネームを右クリックして、「**変更**」をクリックします。「ニックネームの変更」ノートブックがオープンします。
3. 「設定」ページで、追加または除去するすべてのオプションの横にあるチェック・ボックスを選択します。必須指定のオプションを除去することはできません。
4. オプションの値を指定または変更するには、そのオプションで「**値**」フィールドをクリックします。オプションによっては、リストから値を選択するか、クリックして複数の値を選択するか、新規の値を入力するかのいずれかが可能です。
5. 「**OK**」をクリックして、ニックネームを変更し、ノートブックをクローズします。

コマンド行プロンプトからこのタスクを実行するには、`ALTER NICKNAME` ステートメントを使用します。たとえば、以下のようになります。

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

たとえば、表構造ファイル `drugdata1.txt` にニックネーム `DRUGDATA1` が作成されるとします。`CREATE NICKNAME` ステートメントで元々定義されていた完全修飾パスは `/user/pat/drugdata1.txt` でした。

FILE_PATH ニックネーム・オプションを変更するには、以下のステートメントを発行します。

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

関連タスク:

- 575 ページの『ニックネームの変更』

関連資料:

- 576 ページの『ニックネームの変更に関する制限』
- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』

ニックネーム列オプションの変更

列情報は、CREATE NICKNAME および ALTER NICKNAME ステートメントにニックネーム列オプションと呼ばれるパラメーターを使用して指定します。その値は大文字でも小文字でも指定できます。

ニックネーム列オプションは、ALTER NICKNAME ステートメントを使用して、追加、設定、またはドロップすることができます。

前提条件:

ステートメントを発行する許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- SYSADM または DBADM 権限
- ステートメントで指定されたニックネームに対する ALTER 特権
- ステートメントで指定されたニックネームに対する CONTROL 特権
- ニックネームにスキーマ名がある場合は、スキーマに対する ALTERIN 特権
- ニックネームのカatalog・ビューの DEFINER 列に記録された、ニックネームの定義者

制約事項:

『ニックネームの変更に関する制限』のトピックを参照してください。

手順:

列名の変更は、DB2 コントロール・センターまたは DB2 コマンド行プロセッサで行えます。

DB2 コントロール・センターからこのタスクを行うには、以下のようになります。

1. 「ニックネーム」フォルダーを選択します。
2. 変更するニックネームを右クリックして、「変更」をクリックします。「ニックネームの変更」ノートブックがオープンします。
3. 「ニックネーム」ページで、変更する列を選択して「変更」をクリックします。「列の変更」ウィンドウがオープンします。
4. 追加または除去する列オプションを選択します。

5. 追加または変更するオプションについて、オプションの値を指定します。
6. 「OK」をクリックして列オプションを変更し、ウィンドウを閉じます。
7. 「OK」をクリックして、ニックネームを変更し、ノートブックを閉じます。

コマンド行プロンプトからこのタスクを実行するには、ALTER NICKNAME ステートメントを使用します。

例 1: リレーショナル・データ・ソースでの NUMERIC_STRING 列オプションの指定:

NUMERIC_STRING 列オプションは、文字タイプの列 (CHAR および VARCHAR) に適用されます。フェデレーテッド・データベースの照合シーケンスとは異なる照合シーケンスを持つデータ・ソースがあるとして、フェデレーテッド・サーバーは通常、文字データを含む列をデータ・ソース側でソートすることはありません。データはフェデレーテッド・データベースに戻され、ローカルにソートが行われます。しかしここで、列が文字データ・タイプであり、数字 ('0', '1', ..., '9') だけが入っているとします。これは、NUMERIC_STRING 列オプションを 'Y' にすれば示すことができます。こうすることで、DB2 UDB 照会オプティマイザーには、データ・ソース側でソートを実行するオプションが追加されます。ソートをリモート側で実行できれば、データをフェデレーテッド・サーバーでソートするというオーバーヘッドが避けられます。

INDONESIA_SALES という名前の Oracle 表のニックネームが ORA_INDSALES であるとして、表にはデータ・タイプが VARCHAR の POSTAL_CODE という列があります。当初、その列には数字だけが含まれ、NUMERIC_STRING 列オプションは 'Y' に設定されていました。しかし今は、数字と数字以外の文字も入っています。このとき、NUMERIC_STRING 列オプションを 'N' に変更するには、次のステートメントを使用します。

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

例 2: リレーショナル・データ・ソースでの VARCHAR_NO_TRAILING_BLANKS 列オプションの指定:

VARCHAR_NO_TRAILING_BLANKS 列オプションは、末尾ブランクを含まない特定の列を識別するために使用できます。列に対して実行されるすべての操作 (比較演算など) を調べる際に、SQL コンパイラーがこの設定を解析します。

INDONESIA_SALES という名前の Oracle 表のニックネームが ORA_INDSALES であるとして、表にはデータ・タイプが VARCHAR の NAME という列があります。NAME 列には末尾ブランクは含まれていません。このニックネームに VARCHAR_NO_TRAILING_BLANKS オプションを追加するには、次のステートメントを使用します。

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

例 3: 非リレーショナル・データ・ソースでの XPATH 列オプションの指定:

EMPLOYEE は XML データ・ソースのニックネームです。 *fname* 列には XPATH が指定されています。この XPATH 列オプションを別のパスに設定するには、次のステートメントを使用します。

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

関連タスク:

- 575 ページの『ニックネームの変更』

関連資料:

- 576 ページの『ニックネームの変更に関する制限』
- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』

データ・ソース・オブジェクトのローカル・タイプの変更

ニックネームを作成すると、データ・ソース・オブジェクトに関連付けられているデータ・タイプは、フェデレーテッド・データベースに保管されます。データ・ソースには、ラッパーが自動でデータ・タイプを指定するものと、ニックネーム作成時にユーザーがデータ・タイプを指定しなければならないものがあります。

特定のデータ・ソース・オブジェクトの列に、ローカル・タイプを指定することができます。その場合、CREATE TYPE MAPPING ステートメントではなく、ALTER NICKNAME ステートメントを使用します。

重要: 列のローカル・データ・タイプを、対応するリモート・タイプとは大きく異なるタイプに変更すると、結果としてエラーになったり、情報が失われてしまう場合があります。

前提条件:

ステートメントを発行する 許可 ID には、以下の特権が少なくとも 1 つ含まれていなければなりません。

- SYSADM または DBADM 権限
- ステートメントで指定されたニックネームに対する ALTER 特権
- ステートメントで指定されたニックネームに対する CONTROL 特権
- ニックネームにスキーマ名がある場合は、スキーマに対する ALTERIN 特権

ステートメントに関連付けられる許可 ID は、ニックネームのカatalog・ビューの DEFINER 列に記録されているニックネーム定義者でなければなりません。

制約事項:

『ニックネームの変更に関する制限』のトピックを参照してください。

手順:

データ・タイプは、DB2 コントロール・センターまたは DB2 コマンド行から変更できます。

DB2 コントロール・センターからこのタスクを行うには、以下のようになります。

1. 「ニックネーム」フォルダーを選択します。
2. 変更するニックネームを右クリックして、「**変更**」をクリックします。「ニックネームの変更」ノートブックがオープンします。
3. 「ニックネーム」ページで、変更する列を選択して「**変更**」をクリックします。「列の変更」ウィンドウがオープンします。
4. データ・タイプを選択します。
5. 「**OK**」をクリックしてデータ・タイプを変更し、ウィンドウをクローズします。
6. 「**OK**」をクリックして、ニックネームを変更し、ノートブックをクローズします。

コマンド行プロンプトからこのタスクを実行するには、ALTER NICKNAME ステートメントを使用します。たとえば、以下のようになります。

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

文字データを含むローカル列の内容をビット (バイナリー) データとして扱う場合は、ALTER NICKNAME ステートメントに FOR BIT DATA 文節を使用します。この文節を使用して列のローカル・データ・タイプを変更した場合、他のシステムとデータ交換を行ったときには、コード・ページ変換が行われません。リモート・データベースの照合シーケンスに関係なく、比較はバイナリーで行われます。

関連タスク:

- 575 ページの『ニックネームの変更』

関連資料:

- 576 ページの『ニックネームの変更に関する制限』
- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 669 ページの『付録 H. デフォルトの順方向データ・タイプ・マッピング』
- 584 ページの『データ・ソース・オブジェクトのローカル・タイプの変更例』
- 569 ページの『非リレーショナル・データ・ソースでサポートされるデータ・タイプ』

データ・ソース・オブジェクトのローカル・タイプの変更例

以下の例は、データ・ソース・オブジェクトのデータ・タイプを変更する方法を示したものです。

例: 数値データ・タイプ・マッピング:

従業員情報の Oracle 表で、BONUS 列がデータ・タイプ NUMBER(32,3) で定義されているとします。Oracle データ・タイプ NUMBER(32,3) は、デフォルトでは DB2 データ・タイプ DOUBLE (倍精度浮動小数点数データ・タイプ) にマップされます。BONUS 列を含む照会を実行すると、次のような値が戻されます。

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

浮動小数は、小数点を移動する桁数と方向を指示します。この例で、+002 は小数点の位置を右に 2 桁移動することを意味し、+003 は右に 3 桁移動することを意味します。

BONUS 列を含む照会が戻す値を、金額らしくすることもできます。その場合、表の BONUS 列のローカル定義を DOUBLE データ・タイプから DECIMAL データ・タイプに変更します。実際の賞与の形式を反映する精度と位取りを使用します。たとえば、賞与のドル金額 (整数部分) が 6 桁を超えないのであれば、NUMBER(32,3) を DECIMAL(8,2) にマップします。この新しいローカル・タイプの制約によって、BONUS 列を含む照会からは値が次のような形式で戻されます。

```
500.00
1000.00
```

Oracle 表のニックネームは ORASALES です。ORASALES 表の BONUS 列を DB2 DECIMAL (8,2) データ・タイプにマップするには、次の ALTER NICKNAME ステートメントを発行します。

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

ORASALES

Oracle 表に定義したニックネーム。

ALTER COLUMN BONUS

フェデレーテッド・データベース SYSCAT.COLUMNS カタログ・ビューにローカルに定義した列の名前。

LOCAL TYPE DECIMAL(8,2)

列の新しいローカル・タイプであることを表します。

| このマッピングは、ニックネーム ORASALES で識別される Oracle 表の BONUS
| 列にのみ適用されます。BONUS 列が含まれるその他のすべての Oracle データ・
| ソース・オブジェクトは、Oracle NUMBER データ・タイプのデフォルトのデー
| タ・タイプ・マッピングを使用します。

例: 日付データ・タイプ・マッピング:

| Oracle 表 SALES のニックネームは ORASALES です。SALES 表には Oracle
| DATE データ・タイプの列が 1 つ含まれています。デフォルトのタイプ・マッピ
| ングでは、Oracle DATE データ・タイプは DB2 TIMESTAMP データ・タイプにマ
| ップされます。しかし、この列からデータを検索して表示したいのは、日付値だけ
| であるとして。この場合、SALES 表のニックネームを変更し、ローカル・タイプを
| DB2 DATE データ・タイプに変更することができます。

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

例: 非リレーショナル・データ・ソースのデータ・タイプ・マッピング:

| 表構造のファイル drugdata1.txt のニックネームは DRUGDATA1 です。
| drugdata1.txt ファイルには医薬品名をリストした列が含まれており、その列の名
| 前は DRUG です。DRUG 列は当初 CHAR(20) で定義されましたが、列の長さを

| CHAR(30) に変更することが必要になりました。この場合、drugdata1.txt ファイルのニックネームを次のようにして変更し、マッピングを正しい長さに変更することができます。

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

関連タスク:

- 「フェデレーテッド・システム・ガイド」の『データ・タイプ・マッピングの作成』
- 583 ページの『データ・ソース・オブジェクトのローカル・タイプの変更』

関連資料:

- 「SQL リファレンス 第 2 巻」の『ALTER NICKNAME ステートメント』
- 576 ページの『ニックネームの変更に関する制限』

第 29 章 DDL コマンド・リファレンス

この章では、このマニュアルで説明されているラッパー DDL コマンドの構文ステートメント、引き数、およびオプションの詳細を示します。ステートメントは、ラッパーの順に列挙されています。

BioRS DDL 参照情報

CREATE SERVER ステートメント・オプション - BioRS ラッパ ー

BioRS の CREATE SERVER ステートメントのオプションは以下のとおりです。

TYPE サーバー・タイプを指定します。デフォルト値は BioRS です。BioRS ラッパ
ーがサポートする値はデフォルト値のみです。このオプションを指定する
必要はありません。

VERSION

サーバー・バージョンを指定します。デフォルト値は 1.0 です。BioRS ラ
ッパ
ーがサポートする値はデフォルト値のみです。このオプションを指定す
る必要はありません。

NODE BioRS 照会ツールが使用可能なシステムのホスト名を指定します。デフォ
ルト値は *localhost* です。

PORT BioRS サーバーへの接続に使用するポート番号を指定します。デフォ
ルト値は 5014 です。

TIMEOUT

BioRS ラッパ
ーの BioRS サーバーからの応答待機時間を分単位で指定しま
す。デフォルト値は 10 です。このオプションは指定する必要があります。

CASE_SENSITIVE

BioRS サーバーが大文字小文字を区別して名前を扱うかどうかを指定しま
す。有効値は 'Y' または 'N' です。デフォルト値は 'Y' です。

BioRS 製品では、BioRS サーバー・マシンに格納されているデータの
大文字小文字の区別を構成パラメーターで制御します。CASE_SENSITIVE オ
プションは、DB2 Information Integrator における BioRS システム構成パラ
メーターに相当します。BioRS システムと DB2 Information Integrator で、
BioRS サーバーの大文字小文字の区別の構成設定を同期させなければなり
ません。BioRS と DB2 Information Integrator の間で大文字小文字の区別
に関する構成設定を同期させておかないと、DB2 Information Integrator を
介した BioRS データへのアクセスを試行した際にエラーが発生します。

重要: DB2 Information Integrator に新規の BioRS サーバーを作成した
後で、CASE_SENSITIVE オプションを変更または削除することはできま
せん。CASE_SENSITIVE オプションを変更する必要がある場合は、サーバー
全体をドロップしてから再作成する必要があります。BioRS サーバーをドロ

ップした場合、対応する BioRS ニックネームもすべて再作成しなければなりません。DB2 Information Integrator は、ドロップされたサーバーに対応するすべてのニックネームを自動的にドロップします。

関連タスク:

- 82 ページの『BioRS データ・ソースのサーバー定義の登録』
- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 589 ページの『CREATE NICKNAME ステートメント構文 - BioRS ラッパー』

CREATE USER MAPPING ステートメント・オプション - BioRS ラッパー

GUEST

BioRS サーバーの BioRS ゲスト認証メカニズムを使用して運用するかどうかを指定します。有効値は 'Y' または 'N' です。デフォルト値は 'Y' です。

このオプションを 'Y' に設定すると、対象となる DB2 Information Integrator ユーザーはゲストの認証を使用して BioRS サーバーにアクセスします。

このオプションを 'N' に設定すると、対象となる DB2 Information Integrator ユーザーは BioRS サーバーへのアクセスに BioRS 許可 ID とパスワードが必要になります。

ユーザー・マッピングが作成されない場合、またはオプションを指定せずにユーザー・マッピングが作成された場合、DB2 Information Integrator ユーザーはゲスト認証で BioRS サーバーにアクセスします。

REMOTE_AUTHID

DB2 ユーザーの BioRS データ・ソースへのアクセスを可能にするユーザー ID を指定します。このリモート ID は、BioRS アプリケーションで受け入れ可能な形式でなければなりません。このオプションは、GUEST オプションを 'N' に設定した場合に設定する必要があります。

REMOTE_PASSWORD

このリモート ID のパスワードを指定します。このオプションは、GUEST オプションを 'N' に設定した場合に設定する必要があります。

例:

次に示す CREATE USER MAPPING ステートメントは、Biors_Server1 サーバーでユーザー Charlie をユーザー Charlene にマップします。

```
CREATE USER MAPPING FOR Charlie SERVER Biors_Server1
  OPTIONS(GUEST 'N', REMOTE_AUTHID 'Charlene', REMOTE_PASSWORD 'Charlene_pw');
```

関連タスク:

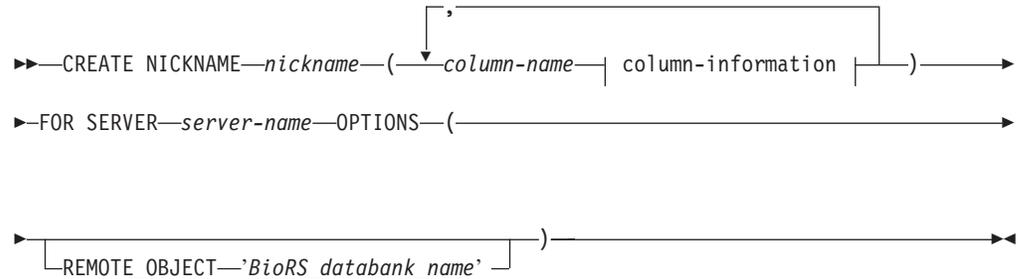
- 83 ページの『BioRS データ・ソースのユーザー・マッピングの登録』

関連資料:

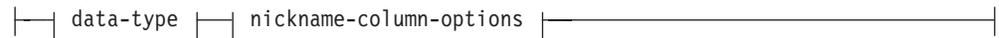
- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』

CREATE NICKNAME ステートメント構文 - BioRS ラッパー

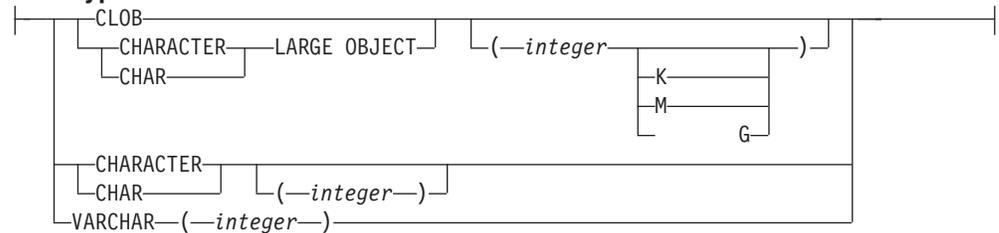
CREATE NICKNAME ステートメントの構文は以下のとおりです。



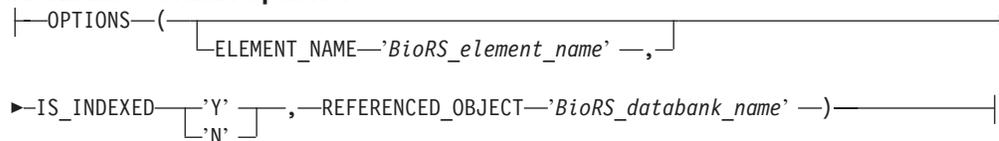
column-information:



data-type:



nickname-column-options:



ニックネーム列オプション

ニックネーム列オプションの値は、単一引用符で囲む必要があります。

ELEMENT_NAME

BioRS エレメント名を指定します。この名前で大文字小文字を区別するかどうかは、BioRS サーバーで大文字小文字を区別するかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。BioRS エレメント名と列名が異なる場合のみ、エレメント名を指定する必要があります。

IS_INDEXED

対応する列が索引付けされているか (述部で列が参照可能か) を示します。有効値は 'Y' と 'N' です。値 'Y' は、BioRS サーバーにより対応するエレメントが索引付けされた列にのみ指定できます。

ニックネームの作成時に、このオプションは値 'Y' により BioRS が索引付けしたエレメントに対応する列に追加されます。

REFERENCED_OBJECT

このオプションは、BioRS データ・タイプが Reference である列にのみ有効です。このオプションは、現在の列が参照する BioRS データ・バンクの名前を指定します。名前の大文字小文字の区別は、BioRS サーバーでの区別および CASE_SENSITIVE サーバー・オプションの値に依存します。

ニックネーム・オプション

ニックネーム・オプション値は、単一引用符で囲む必要があります。

REMOTE_OBJECT

ニックネームに関連する BioRS データ・バンク名を指定します。この名前によって、ニックネームのスキーマと BioRS データ・バンクが決まります。この名前によって、そのニックネームと他のニックネームの関係も指定されます。この名前で大文字小文字を区別するかどうかは、BioRS サーバーで大文字小文字を区別するかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。

重要: ALTER NICKNAME ステートメントではこの名前を変更または削除できません。このオプションで使用される BioRS データ・バンクの名前を変更した場合、ニックネーム全体を削除してから、再びそれを再作成しなければなりません。

関連タスク:

- 84 ページの『BioRS データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 86 ページの『CREATE NICKNAME ステートメント - BioRS ラッパーの例』

BLAST DDL 参照情報

CREATE SERVER ステートメントの引き数 - BLAST ラッパー

BLAST ラッパーの CREATE SERVER 引き数:

TYPE 所定のサーバーを使用して実行される BLAST 検索のタイプを決定します。この引き数は必須です。これは、blastn、blastp、blastx、tblastn、tblastx のうちのいずれかの値に設定されなければなりません。

VERSION

使用しようとするサーバーのバージョンを指定します。これは、実行している blastall のバージョンに設定する必要があります。この引き数は必須です。

WRAPPER

CREATE WRAPPER ステートメントを使用して登録したラッパー名を指定します。この引き数は必須です。

BLAST ラッパーのサーバー・オプション:

CREATE SERVER ステートメントで指定できる BLAST のオプションは、以下のとおりです。

- DAEMON_PORT
- NODE
- USE_CLOB_SEQUENCE

関連タスク:

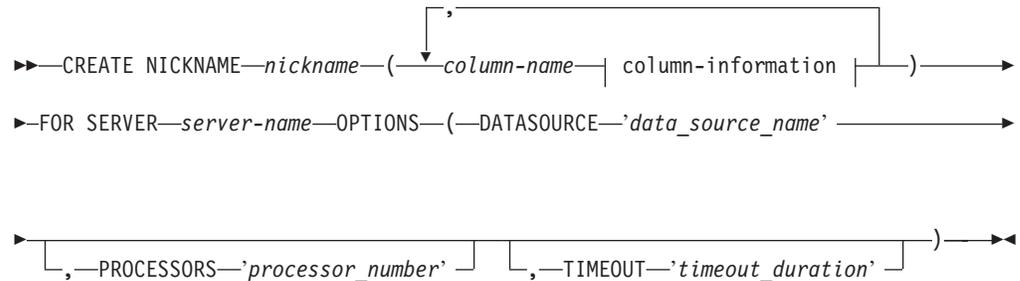
- 121 ページの『BLAST データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』

CREATE NICKNAME ステートメント構文 - BLAST ラッパー

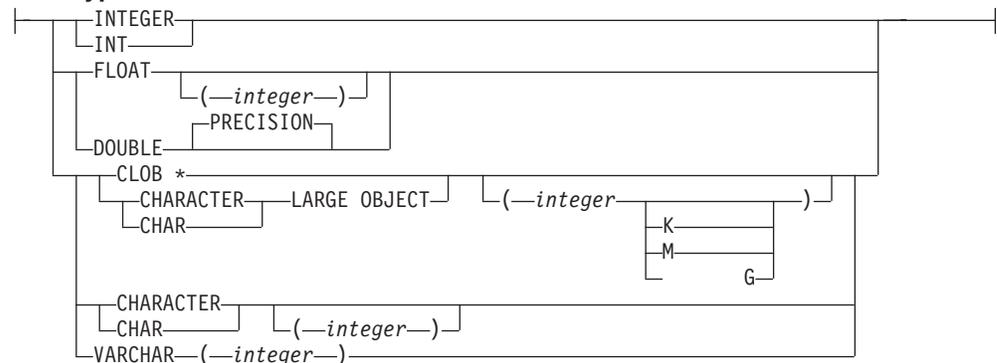
CREATE NICKNAME ステートメントの構文は以下のとおりです。



column-information:



data-type:



column-option:



nickname-column-options:

```

|---OPTIONS---(---INDEX---'index_number' ---,---DELIMITER---'delimiter' ---)
|
|---DEFAULT---'new_default_value' ---|

```

制約事項: BLAST ラッパーでは、CLOB の長さは 5 メガバイト (5MB) に制限されています。

BLAST データ・ソースで CREATE NICKNAME ステートメントに指定できるオプションには、以下の 2 つのタイプがあります。

- ニックネーム列オプション
- ニックネーム・オプション

関連タスク:

- 121 ページの『BLAST データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』
- 661 ページの『付録 G. フェデレーテッド・システムのニックネーム列オプション』
- 649 ページの『付録 F. フェデレーテッド・システムのニックネーム・オプション』
- 127 ページの『CREATE NICKNAME ステートメント - BLAST ラッパーの例』

Documentum DDL 参照情報

CREATE SERVER ステートメントの引き数とオプション - Documentum ラッパー

Documentum の CREATE SERVER ステートメントに関連した引き数

TYPE データ・ソースのタイプを指定します。 Documentum の場合、タイプは DCTM になります。この引き数は必須です。

VERSION

データ・ソースのバージョンを指定します。 EDMS98 の場合、値は '3' になります。 4i の場合、値は '4' になります。この引き数は必須です。

WRAPPER

このサーバーと関連したラッパーの名前を指定します。この引き数は必須です。

Documentum の CREATE SERVER ステートメントに関連したオプション

CONTENT_DIR

GET_FILE、GET_FILE_DEL、GET_RENDITION、および GET_RENDITION_DEL 疑似列によって検索された内容ファイルを保管するための、ローカルにアクセス可能なルート・ディレクトリーの名前を指定し

ます。これは、これらの疑似列を使用できるすべてのユーザーが書き込み可能でなければなりません。そのデフォルト値は /tmp です。このオプションは任意です。

NODE Documentum Docbase の実際の名前を指定します。このオプションは必須です。

OS_TYPE

Docbase サーバーのオペレーティング・システムを指定します。有効な値は AIX、SOLARIS、および WINDOWS です。このオプションは必須です。

RDBMS_TYPE

Docbase で使用される RDBMS を指定します。有効な値は DB2、INFORMIX、ORACLE、SQLSERVER、または SYBASE です。このオプションは必須です。

TRANSACTIONS

サーバー・トランザクション・モードを指定します。有効な値は以下のとおりです。

- NONE — トランザクションは使用可能になりません。
- QUERY — トランザクションは Dctm_Query メソッドでのみ使用可能です。
- ALL — トランザクションは Dctm_Query メソッドで使用可能です。このリリースでは、ALL と QUERY の機能は同じです。

デフォルトは QUERY です。このオプションは任意です。

関連タスク:

- 195 ページの『Documentum データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

CREATE USER MAPPING ステートメントのオプション - Documentum ラッパー

オプションの定義

REMOTE_AUTHID

リモート・サーバーにおける許可 ID。

REMOTE_PASSWORD

リモート・サーバーにおけるパスワード。

REMOTE_DOMAIN

リモート・サーバーにおける Windows ネットワーキング・ドメイン。
Windows プラットフォームの場合のみ有効です。

関連タスク:

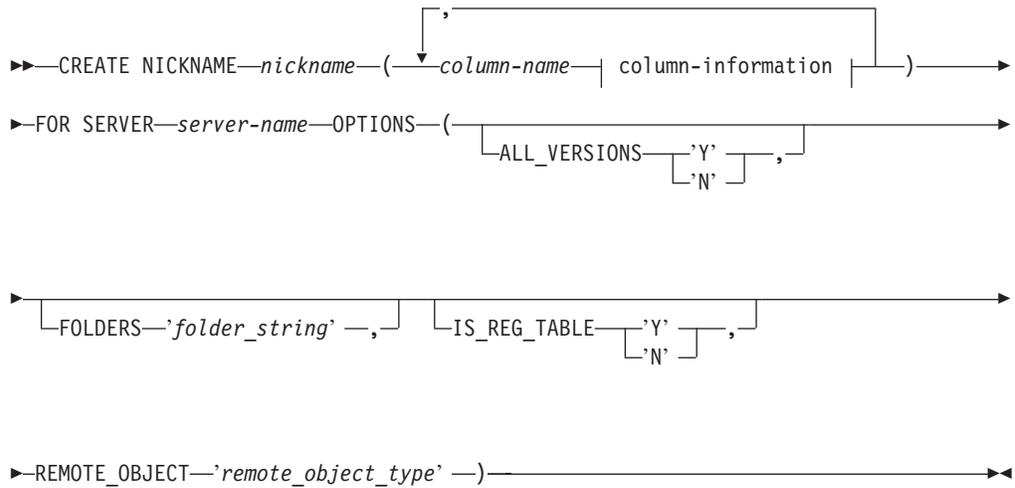
- 196 ページの『Documentum データ・ソースのユーザー・マッピングの登録』

関連資料:

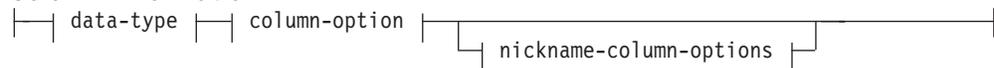
- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』

CREATE NICKNAME ステートメント構文 - Documentum ラッパー

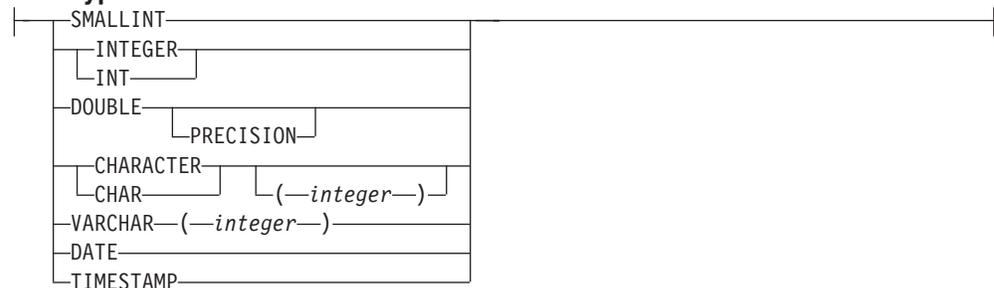
Documentum の CREATE NICKNAME ステートメントの構文は以下のとおりです。



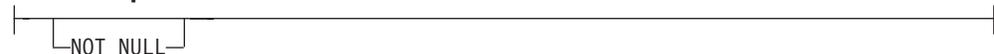
column-information:



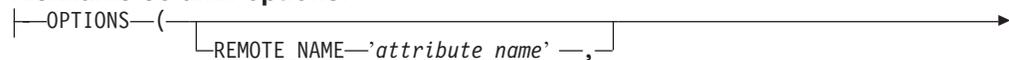
data-type:

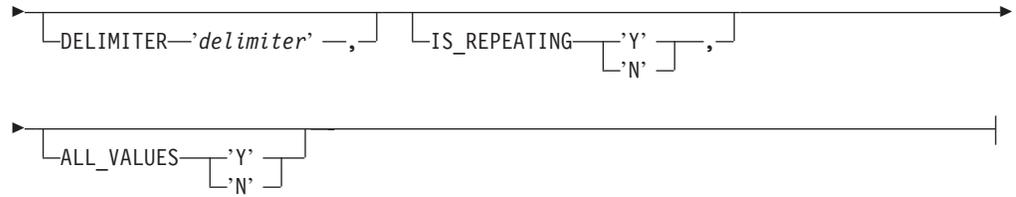


column-option:



nickname-column-options:





以下は Documentum の CREATE NICKNAME ステートメントに関連した列オプションです。

NOT NULL

TIMESTAMP および DATE と定義されたものを除くすべての単一値の列は、NOT NULL として定義しなければなりません。反復する属性は、ニックネームで NOT NULL として定義してはなりません。

以下は Documentum の CREATE NICKNAME ステートメントに関連したニックネーム列オプションです。

ニックネーム列オプションの値は、単一引用符で囲む必要があります。

ALL_VALUES

繰り返し属性のすべての値を、指定された区切り文字で分離して戻すことを指定します。このオプションが指定されていないか、'N' の場合、繰り返し属性の最後の値のみが戻されます。DELIMITER の箇所で説明されているように、ALL_VALUES は IS_REPEATING オプションが 'Y' である VARCHAR 列に対してのみ指定できます (IS_REG_TABLE = 'Y' の場合は無効です)。

DELIMITER

反復する属性の複数を連結するときに使用する区切り文字を指定します。区切り文字は 1 文字以上です。デフォルトの区切り文字はコンマです。このオプションは、IS_REPEATING オプションが 'Y' に設定されているデータ・タイプ VARCHAR のオブジェクトの属性に対してのみ有効です。このオプションは任意です。

IS_REPEATING

列が複数值であることを示します。有効な値は 'Y' および 'N' です。デフォルトは 'N' です。このオプションは任意です。

以下の場合、最後の値のみが戻されます。

- 非 VARCHAR 繰り返し属性
- ALL_VALUES 'N' が指定されているときの VARCHAR 列

この制限に対処するため、繰り返し属性列に二重の定義を作成できます。

REMOTE_NAME

対応する Documentum 属性または列の名前を指定します。このオプションは、リモート属性または列名をローカルの DB2 列名にマップします。これはデフォルトの DB2 列名になります。このオプションは任意です。

以下は Documentum の CREATE NICKNAME ステートメントに関連したニックネーム列オプションです。

ニックネーム・オプション値は、単一引用符で囲む必要があります。

ALL_VERSIONS

すべてのオブジェクト・バージョンを検索するかどうか指定します。有効な値は 'y'、'Y'、'n'、および 'N' です。デフォルト値の 'N' は、現行オブジェクト・バージョンのみが照会プロセスに組み込まれることを意味します。このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。このオプションは任意です。

FOLDERS

1 つ以上の論理的に結合され、構文的に正確な Documentum FOLDER 述部を含むストリングを指定します。FOLDER 述部を指定すると、このニックネームによって表現される文書のセットを、指定したフォルダー内のものに制限します。

このオプションを指定するときは、FOLDERS オプションの値全体を単一引用符で囲み、ストリング内では単一引用符の代わりに二重引用符を使用します。

たとえば、次のものを挿入するとします。

```
FOLDER('/Tools',DESCEND) OR FOLDER('/Cars')
```

次の FOLDERS オプションを指定します。

```
FOLDERS 'FOLDER("/Tools",DESCEND) OR FOLDER("/Cars")'
```

このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。このオプションは任意です。

IS_REG_TABLE

REMOTE_OBJECT オプションによって指定されたオブジェクトが Documentum の登録済みの表かどうかを指定します。有効な値は 'y'、'Y'、'n'、および 'N' です。デフォルト値は 'N' です。このオプションは任意です。

ALTER NICKNAME ステートメントでこのオプションを変更して、ニックネームを Documentum オブジェクトから登録済みの表に (またはその逆に) 変更することはできません。そうする代わりに、ニックネームを DROP してもう一度 CREATE する必要があります。

REMOTE_OBJECT

ニックネームに関連した Documentum オブジェクト・タイプの名前を指定します。名前は任意の Documentum オブジェクト・タイプまたは登録済みの表にできます。登録済みの表になる場合、表の所有者の名前を接頭部に付ける必要があります。登録済みの表が Docbase 所有者に所属する場合、dm_dbo を所有者の名前に使用できます。このオプションは必須です。

新規オプションの構造がオリジナルのオプションの構造に類似していなければ、ALTER NICKNAME を使用して REMOTE_OBJECT オプションの値を変更するとエラーになります。

関連タスク:

- 197 ページの『Documentum データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE NICKNAME ステートメント』
- 202 ページの『CREATE NICKNAME ステートメント - Documentum ラッパーの例』

Excel DDL 参照情報

CREATE SERVER ステートメントの引き数 - Excel ラッパー

Excel の CREATE SERVER ステートメントに関連した引き数

WRAPPER

関連する CREATE WRAPPER ステートメントに登録したラッパー名を指定します。この引き数は必須です。

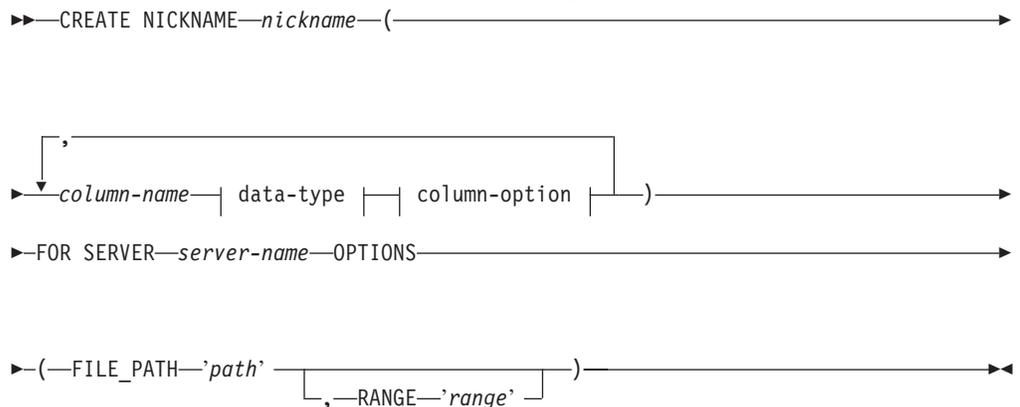
関連タスク:

- 247 ページの『Excel データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE SERVER ステートメント』

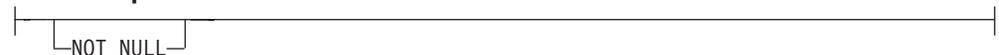
CREATE NICKNAME ステートメント構文 - Excel ラッパー



data-type:



column-option:



ここで、

FOR SERVER

関連する CREATE SERVER ステートメントに登録したサーバーを示します。このサーバーを使用して、Excel スプレッドシートにアクセスします。サーバー名を指定します。

以下のリストは、Excel の CREATE NICKNAME オプションの説明です。

FILE_PATH

アクセスする Excel スプレッドシートの完全修飾ディレクトリー・パスおよびファイル名を指定します。

データ・タイプはそれぞれの列の中で一貫している必要があります。列データ・タイプはニックネームの登録プロセス中に正確に記述しなければなりません。

Excel ラッパーは、Excel ワークブック内で基本スプレッドシートにのみアクセスできます。

スプレッドシート内のブランク・セルは、NULL として解釈されます。

スプレッドシート内には最大で連続 10 個のブランクを含めることができ、データ・セット内に含まれます。10 個を超える連続ブランクは、データ・セットの終了として解釈されます。

スプレッドシートに、ブランク列を含めることができます。しかし、これらの列は、たとえそれが使用されない場合でも、有効なフィールドとして登録され、記述されなければなりません。

データベースのコード・ページはファイルの文字セットと一致する必要があります。そうでない場合、予期しない結果が生じる場合があります。

RANGE

データ・ソースで使用されるセルの範囲を指定します。このオプションは必須ではありません。

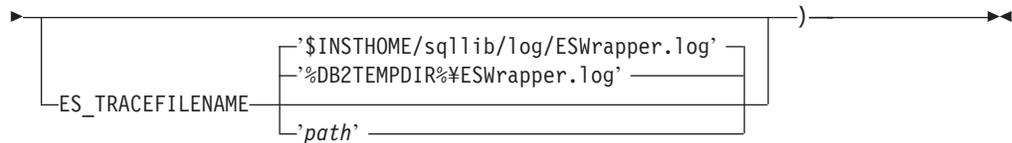
範囲オプション値の構文または意味エラーはどれも SQL1882E メッセージとなります。次のようなエラーがあります。

- 左上と右下の標識が正しい方向になっていない。方向が正しくないのは、左上セルの標識が右下セル標識の下または右にある場合です。
- 範囲値によって指定された列の数が CREATE NICKNAME ステートメントで指定された列の数に対応しない。
- 無効文字または他の構文エラーが見つかった。

以下は、RANGE ニックネーム・オプションの例です。

```
CREATE NICKNAME excel2
(c1 VARCHAR (10),
c2 VARCHAR (10),
c3 VARCHAR (10),
c4 VARCHAR (10)
) FOR SERVER excel_server
OPTIONS (FILE_PATH 'C:¥My Documents¥test2.xls',
RANGE 'B2:E5');
```

この例では、**B2** はセル範囲の左上を表し、**E5** はセル範囲の右下を表します。B2 指定の中にある文字 **B** は、列仕様です。B2 表記の中にある数 **2** は行番号です。



SERVER

このサーバー定義の固有名を指定します。このパラメーターは必須です。

WRAPPER

このサーバー定義で使用したいすでに登録済みの Extended Search ラッパーの名前を指定します。このパラメーターは必須です。

ES_HOST

検索する Extended Search サーバーの完全修飾ホスト名または IP アドレスを指定します。このオプションは必須です。

ES_PORT

この Extended Search が要求を listen するポート番号を指定します。このオプションを省略する場合、デフォルト値は 6001 です。

ES_TRACING

リモート Extended Search サーバーが作成するエラー・メッセージ、警告メッセージ、および通知メッセージに関して、トレースを有効にするかどうかを指定します。デフォルト値の OFF は、トレース・メッセージがログに記録されないことを表します。

ES_TRACELEVEL

トレースが有効になっている場合、このオプションはログ・ファイルに書き込まれるメッセージのタイプを指定します。デフォルト値の C は、重大メッセージだけをログに記録します。以下のトレース・レベルを、個別に有効および無効にできます。

- C – 重大なエラー・メッセージ
- N – 重大ではないメッセージ
- W – 警告メッセージ
- I – 通知メッセージ

例:

```

ES_TRACELEVEL 'W'
ES_TRACELEVEL 'CN'
ES_TRACELEVEL 'CNWI'
  
```

ES_TRACEFILENAME

トレースが有効になっている場合、このオプションはメッセージが書き込まれるディレクトリーおよびファイルの名前を指定します。このオプションを省略した場合、

- UNIX では、デフォルト値は \$INSTHOME/sql1lib/log/ESWrapper.log です。
- Windows では、デフォルト値は %DB2TEMPDIR%ESWrapper.log です。

関連タスク:

- 266 ページの『Extended Search データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE SERVER ステートメント』

CREATE USER MAPPING ステートメント構文 - Extended Search ラッパー

```
▶▶ CREATE USER MAPPING FOR authorization-name SERVER server-name OPTIONS
▶ ( REMOTE_AUTHID 'user-id' , REMOTE_PASSWORD 'password' )
```

FOR

Extended Search データ・ソースへのアクセスを許可したい DB2 ユーザーのユーザー ID を指定します。このパラメーターは必須です。

SERVER

ユーザーが検索する Extended Search サーバー用に作成されたすでに登録済みのサーバー定義の名前を指定します。このパラメーターは必須です。

REMOTE_AUTHID

この DB2 ユーザーが Extended Search データ・ソースにアクセスできるようにするユーザー ID を指定します。このリモート ID は、検索されるデータ・ソースで必要なフォーマットでなければなりません。このオプションは必須です。

REMOTE_PASSWORD

このリモート ID のパスワードを指定します。このオプションは必須です。

関連タスク:

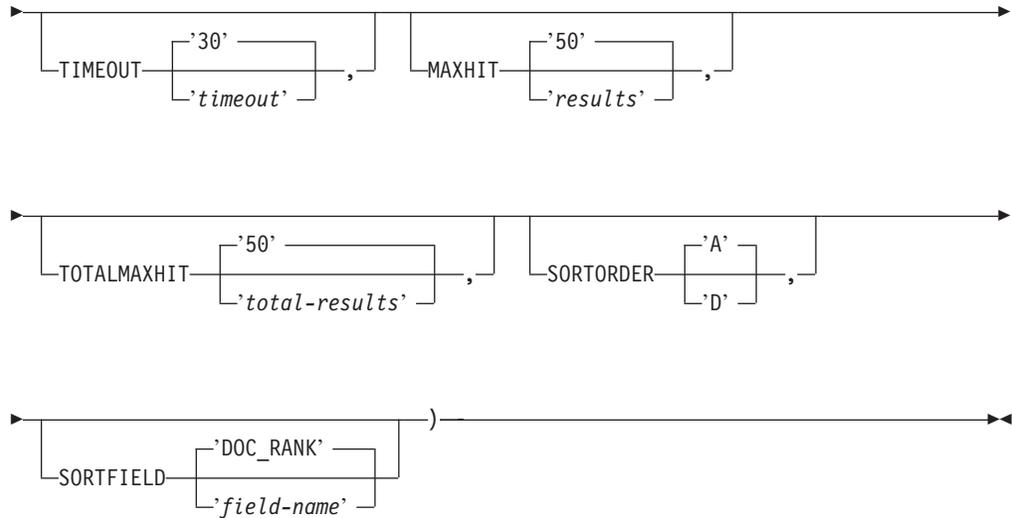
- 267 ページの『Extended Search データ・ソースのユーザー・マッピングの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE USER MAPPING ステートメント』

CREATE NICKNAME ステートメント構文 - Extended Search ラッパー

```
▶▶ CREATE NICKNAME nickname ( column-name data-type )
▶ FOR SERVER server-name OPTIONS ( APPLICATIONID 'application-id' ,
▶ CATEGORY 'category-name' ,
▶ DATASOURCE 'source-name' ,
▶ VERTICAL_TABLE [ 'NO' ] [ 'YES' ] )
```



data-type:



NICKNAME

この Extended Search ニックネーム表の固有な名前を指定します。この名前は、定義されるスキーマ内の他のすべてのニックネームと区別される必要があります。このパラメーターは必須です。

column-name

1 つ以上のユーザー定義列名を指定します。列名は、Extended Search 構成データベースで定義されるネイティブまたはマップ・フィールドの名前に一致する必要があります。このパラメーターはオプションです。

data-type

指定された列の SQL データ・タイプを指定します。このデータ・タイプは、Extended Search 構成データベースのこのフィールドに対して定義されるデータ・タイプに対応している必要があります。たとえば、String データ・タイプの Extended Search データ・ソースのフィールドを検索するには、ニックネーム表にあるこのフィールドの VARCHAR 列を定義します。column-name を指定するには、このパラメーターが必要です。

FOR SERVER

検索したい Extended Search サーバー用に作成されたすでに登録済みのサーバー定義の名前を指定します。このパラメーターは必須です。

APPLICATIONID

検索する Extended Search アプリケーションの名前を指定します。この名前は、Extended Search 構成データベースに存在していなければなりません。このパラメーターは必須です。

CATEGORY

検索する 1 つ以上の Extended Search カテゴリーを指定します。このオプションを省略する場合、少なくとも 1 つのデータ・ソース名を指定しなければなりません。複数のカテゴリーを指定するには、カテゴリー名をセミコロンで区切ります。例:

```
CATEGORY 'LotusNotes;MSAccess;LDAP'
```

DATASOURCE

検索したい 1 つ以上の Extended Search データ・ソースを指定します。このオプションを省略する場合、少なくとも 1 つのカテゴリー名を指定しなければなりません。複数のデータ・ソースを指定するには、データ・ソース名をセミコロンで区切ります。例:

```
DATASOURCE 'AltaVista;Google!;CNN'
```

VERTICAL_TABLE

検索結果の表示フォーマットを指定します。YES を指定すると、Extended Search は、単なるユーザー定義列ではなく、戻りが可能であるとして構成されているすべてのフィールドを戻します。ラッパーは列名の垂直リストとして結果をニックネーム表に保管します。デフォルト値は NO です。

TIMEOUT

要求されたタイムアウトになるまでサーバーからの応答を待つ秒数を指定する INTEGER です。このオプションは任意です。デフォルト値は 30 です。

MAXHIT

検索されている各ソースから戻される結果の最大数を指定する INTEGER です。このオプションは任意です。デフォルト値は 50 です。

TOTALMAXHIT

検索されているすべてのソースから戻される結果の最大数を指定する INTEGER です。ラッパーはそれらの結果を 1 つの結果セットに結合します。このオプションは任意です。デフォルト値は 50 です。

SORTORDER

検索結果の戻りのソート順序 (昇順 (A) か降順 (D のいずれか) を指定します。デフォルト値は A です。

SORTFIELD

検索結果がソートされなければならないフィールドの名前を指定します。デフォルト値の DOC_RANK は、Extended Search が検索文書の関連性を評価するのに使用するフィールドです。別のフィールド名を指定する場合、検索するソースにその名前が存在することを確認してください。

関連概念:

- 260 ページの『Extended Search のニックネーム』
- 261 ページの『Extended Search の垂直表』

関連タスク:

- 268 ページの『Extended Search データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE NICKNAME ステートメント』
- 272 ページの『Extended Search ラッパー - 照会の例』

CREATE FUNCTION ステートメント構文 - Extended Search ラッパー

CREATE FUNCTION ステートメントの構文は以下のとおりです。

```
▶▶CREATE FUNCTION ESWRAPPER.ES_SEARCH(—INTEGER—,—VARCHAR(1024)—)—————▶▶
▶—RETURNS INTEGER AS TEMPLATE DETERMINISTIC NO EXTERNAL ACTION—————▶▶
```

INTEGER

照会参照パラメーターを定義します。照会では、このパラメーターはこのカスタム関数が呼び出されるニックネーム表で定義される INTEGER 列の名前を指定する必要があります。この値は、ニックネームのバインド列である必要があり、定数 (例: DOC_RANK) であってはなりません。

参照パラメーターは、ES_SEARCH 関数を適用すべきニックネームを識別します。パラメーターそのものは評価されません。

SELECT ステートメントが複数の表を FROM 文節に含み、WHERE 文節が ES_SEARCH ステートメントを含む場合、参照パラメーターによって特定の検索ステートメントが属する表を DB2 に知らせることができます。例:

```
SELECT *
FROM   ES_Nickname_1 as N1, ES_Nickname_2 as N2
WHERE  ESWRAPPER.ES_SEARCH(N1.DOC_RANK, 'IBM')=1 AND
      ESWRAPPER.ES_SEARCH(N2.DOC_RANK, 'LOTUS')=1
```

VARCHAR(1024)

照会式を定義します。照会では、このパラメーターは Extended Search 汎用照会言語を使用する文字列を指定する必要があります。

関連タスク:

- 269 ページの『Extended Search ラッパーのカスタム関数の登録』

関連資料:

- 「SQL リファレンス 第2巻」の『CREATE FUNCTION (ソースまたはプレート) ステートメント』
- 272 ページの『Extended Search ラッパー - 照会の例』
- 274 ページの『Extended Search ラッパー - 汎用照会言語』

Entrez DDL 参照情報

CREATE SERVER ステートメントの引き数 - Entrez ラッパー

Entrez の CREATE SERVER ステートメントの引き数

TYPE データ・ソースのタイプを指定します。サーバー・タイプの許容値は、PubMed および Nucleotide です。これらは大文字小文字を区別しません。

VERSION

使用しようとする NCBI XML スキーマのバージョンを指定します。この引き数はオプションです。サーバーのバージョンを指定しない場合、デフォルトは 1.0 です。

WRAPPER

CREATE WRAPPER ステートメントを使用して登録したラッパー名を指定します。

関連タスク:

- 222 ページの『Entrez データ・ソースのサーバーの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』
- 629 ページの『付録 D. フェデレーテッド・システムのサーバー・オプション』

CREATE NICKNAME ステートメントのオプション - Entrez ラッパー

以下のリストは、Entrez の CREATE NICKNAME オプションの説明です。

REMOTE_OBJECT

ニックネームに関連した Entrez オブジェクト・タイプの名前を指定します。この名前によって、ニックネームのスキーマと NCBI データベース、および他のニックネームとの関係が決定されます。この名前は大文字小文字が区別されません。

PARENT

REMOTE_OBJECT オプションによって親の名前変更が行われた子ニックネームに対してのみ指定されます。PARENT オプションは、DB2 スキーマ内で複数のニックネーム・ファミリーが定義されている場合に、子を親に関連付けます。この名前は大文字小文字が区別されます。

関連タスク:

- 224 ページの『Entrez データ・ソースのニックネームの登録』

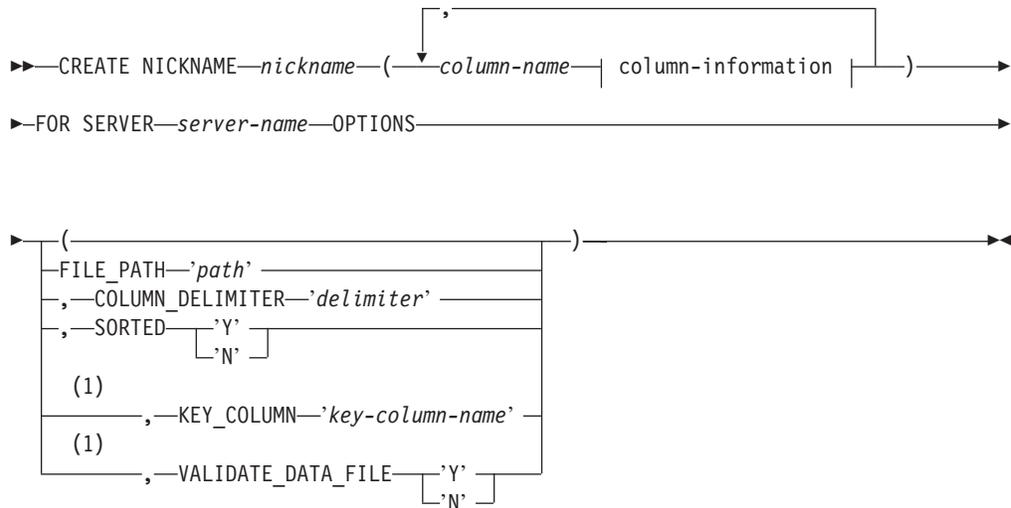
関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

表構造ファイル DDL 参照情報

CREATE NICKNAME ステートメント構文 - 表構造ファイル・ラッパー

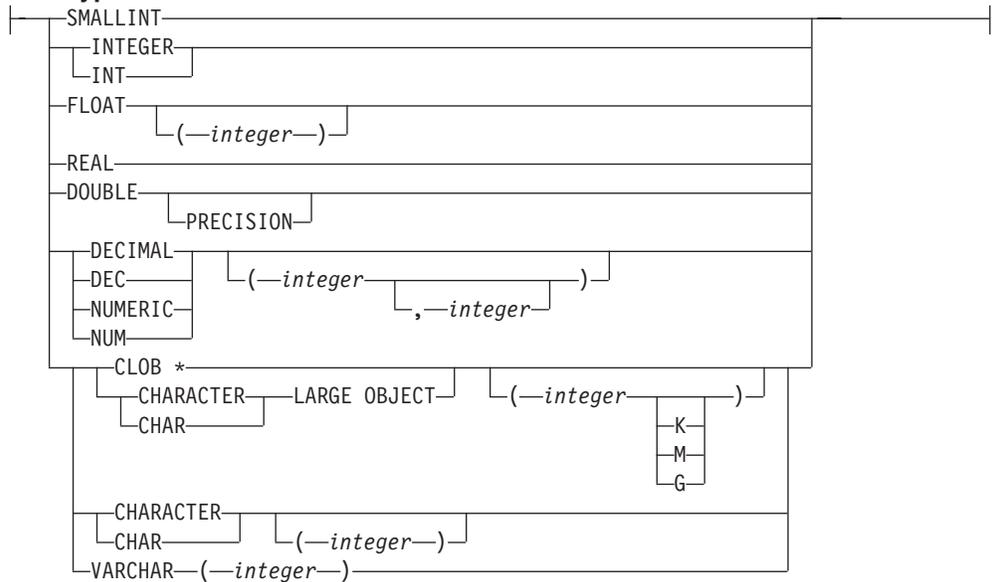
CREATE NICKNAME ステートメントの構文は以下のとおりです。



column-information:



data-type:



column-option:



nickname-column-options:



注:

- 1 ソートされていないファイルには許可されません。ソートされたファイルについてはオプションです。

制約事項: 表構造ファイル・ラッパーの場合、CLOB の長さは、5 メガバイト (5MB) までです。

nickname

アクセスされる表構造のファイルの固有なニックネーム。これは、登録されるスキーマ内の他のすべてのニックネーム、表、およびビューと区別される必要があります。

column-name

表構造のファイルの各フィールドに指定される固有名。各列名にそのデータ・タイプが続きます。サポートされる列タイプは、CHAR、CLOB、DECIMAL、DOUBLE、FLOAT、INTEGER、REAL、SMALLINT、および VARCHAR です。

CHARACTER(*integer*) または **CHAR(*integer*)** または **CHARACTER** または **CHAR**
長さ *integer* の固定長文字ストリング。1 ~ 254 の範囲にすることができます。長さの指定がない場合は、1 文字の長さを指定したものと見なされます。

CLOB(*integer*)

長さ *integer* の文字ラージ・オブジェクト。最大長は 5 MB です。長さの指定がない場合は、1 MB の長さを指定したものと見なされます。

DECIMAL(*precision-integer*, *scale-integer*) または **DEC(*precision-integer*, *scale-integer*)**

10 進数。

最初の整数は数値の精度です。つまり、桁の合計数です。この値は、1 ~ 31 の範囲です。

2 番目の整数は数値の位取りです。つまり、小数点の右側の桁数です。この値は、0 から数値の精度の範囲となります。

精度と位取りが指定されない場合、5,0 のデフォルト値が使用されます。

NUMERIC および **NUM** という語は、**DECIMAL** および **DEC** の同義語として使用できます。

DOUBLE または **DOUBLE PRECISION**

倍精度浮動小数点。

FLOAT(*integer*)

integer の値に応じて、短精度か倍精度の浮動小数点数となります。 *integer* の値は、1 ~ 53 となります。1 ~ 24 の値は短精度を示し、25 ~ 53 の値は倍精度を示します。

INTEGER または **INT**

長精度整数。

REAL 単精度浮動小数点。

SMALLINT

短整数。

VARCHAR(integer)

最大長 *integer* の可変長文字ストリング。1 ~ 32672 の範囲にすることができます。

NOT NULL

列に NULL 値が入るのを防止します。

ラッパーは NOT NULL 制約を強制しませんが、DB2 は強制します。ニックネームを作成して、列に NOT NULL 制約を付け、列に NULL 値の入った行を選択する場合、DB2 は NULL 値を NOT NULL 列に割り当てることはできないことを述べる SQL0407N エラーを出します。

この規則の例外はソートされたニックネームの場合です。ソートされたニックネームのキー列を NULL にすることはできません。ソートされたニックネームに関して NULL キー列が検出されると、キー列が欠落しているという SQL1822N エラーが出ます。

FOR SERVER

CREATE SERVER ステートメントを使用してすでに登録したサーバーを指定します。このサーバーは、表構造ファイルにアクセスのに使用されます。

FILE_PATH

アクセスされる表構造のファイルへの完全修飾パス (単一引用符で囲まれている)。データ・ファイルは標準ファイルまたはシンボリック・リンクでなければならず、パイプや別の非標準ファイル・タイプにすることはできません。FILE_PATH または DOCUMENT ニックネーム列オプションのいずれかを指定しなければなりません。FILE_PATH ニックネーム・オプションが指定されている場合には、DOCUMENT ニックネーム列オプションは指定できません。

SORTED

データ・ソース・ファイルがソートされるか、ソートされないかを指定します。このオプションは 'Y'、'y'、'n'、または 'N' のいずれも受け入れます。デフォルト値は 'N' です。

ソートされるデータ・ソースは、LC_COLLATE 各国語サポート・カテゴリーの設定によって定義されている現在のロケールの照合順序に従って、昇順でソートされなければなりません。

データ・ソースがソートされていることを指定する場合、VALIDATE_DATA_FILE を 'Y' に設定することをお勧めします。

COLUMN_DELIMITER

表構造のファイルの列を区切るのに使用される区切り文字 (単一引用符で囲まれている)。区切り文字は、1 つ以上の文字になります。列区切り文字が定義されていない場合、列区切り文字はデフォルトのコンマになります。単一引用符は区切り文字としては使用できません。列区切り文字は、ファイル全体で一貫性がなければなりません。NULL 値は 2 つの区切り文字が続くことによって表されます。NULL フィールドが行の末尾にある場合、区切り文字に行終止符を続けることによって表されます。列区切り文字は、列の有効なデータとして存在することはできません。たとえば、列の 1 つにコンマが組み込まれたデータが入っている場合、列区切り文字としてコンマを使用することはできません。

KEY_COLUMN

単一引用符で囲まれている、ファイルをソートするキーとなるファイル内の列の名前。このオプションはソート・ファイルでのみ使用してください。DOCUMENT ニックネーム列オプションで指定された列は、キー列として指定してはなりません。

単一系列のキーのみがサポートされます。複数列のキーは許可されていません。値は、CREATE NICKNAME ステートメントに定義されている列の名前にする必要があります。列は必ず昇順でソートされます。ソート・ニックネームに値が指定されていない場合、デフォルトとしてこれはニックネームが付けられたファイルの最初の列となります。NOT NULL オプションをニックネーム・ステートメントの中のその定義に追加することによって、キー列が、NULL 不可能を指定されることをお勧めします。

このオプションには大文字小文字の区別があります。ただし DB2 は、列が二重引用符によって定義されていない限り、列名を大文字に変換します。

VALIDATE_DATA_FILE

ソートされるファイルの場合、このオプションは、キー列が昇順でソートされていることをラッパーがチェックするかどうか、および NULL キーをチェックするかどうかを指定します。このオプションに有効な値は 'Y' または 'N' だけ (単一引用符で囲まれる) です。このチェックは、登録時に一度だけ行われます。このオプションが指定されない場合、妥当性検査は行われません。DOCUMENT ニックネーム列オプションがファイル・パスに使用されている場合には、このオプションは使用できません。

DOCUMENT

表構造ファイルの種類を指定します。現在、このラッパーはこのオプションに FILE のみをサポートしています。DOCUMENT オプションでは、ニックネーム当たり 1 つの列しか指定できません。DOCUMENT オプションに関連付けられる列は、データ・タイプ VARCHAR または CHAR の列でなければなりません。

FILE_PATH ニックネーム・オプションではなく、DOCUMENT ニックネーム列オプションを使用するということは、照会の実行中にこのニックネームに対応するファイルが提供されることを意味します。DOCUMENT オプションが「FILE」値をもっている場合、照会の実行中に提供されるものは、このニックネームのニックネーム定義に一致するスキーマを持つファイルの絶対パスであることを意味します。以下の CREATE NICKNAME の例は、DOCUMENT ニックネーム列オプションの使用例を示しています。

```
CREATE NICKNAME customers
(
  doc VARCHAR(100) OPTIONS(DOCUMENT 'FILE'),
  name VARCHAR(16),
  address VARCHAR(30),
  id VARCHAR(16)
)
FOR SERVER file_server
```

以下の照会は (WHERE 文節で 表構造ファイルのロケーションを指定している) customers ニックネームで実行することができるようになります。

```
SELECT name, address, id FROM customers
WHERE doc='/home/db2user/Customers.txt'
```

関連タスク:

- 402 ページの『表構造ファイルのサーバーの登録』

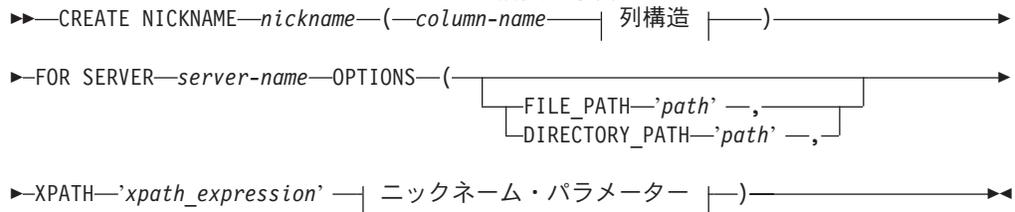
関連資料:

- 「SQL リファレンス 第2巻」の『CREATE NICKNAME ステートメント』
- 403 ページの『CREATE NICKNAME ステートメント - 表構造ファイル・ラッパーの例』

XML DDL 参照情報

CREATE NICKNAME ステートメント構文 - XML ラッパー

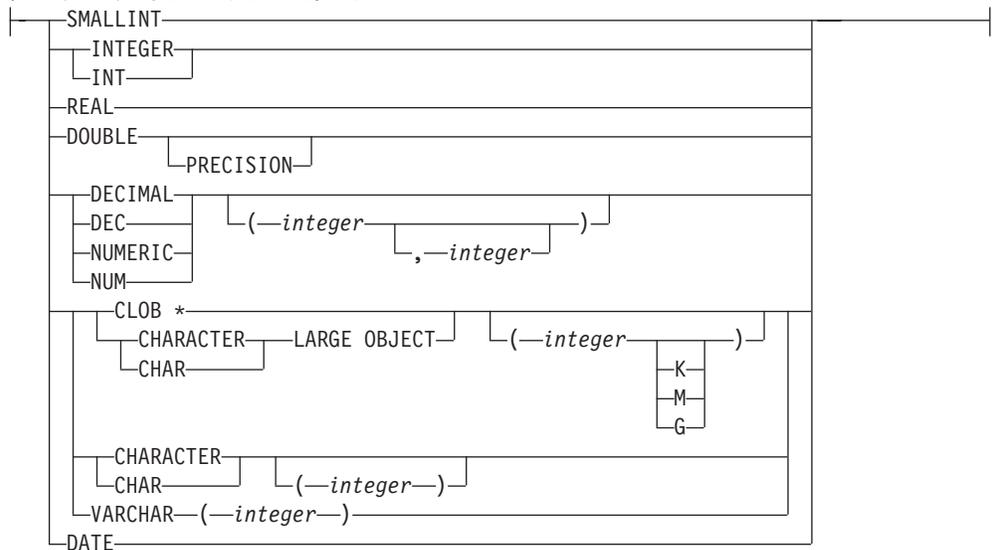
CREATE NICKNAME ステートメントの構文は以下のとおりです。



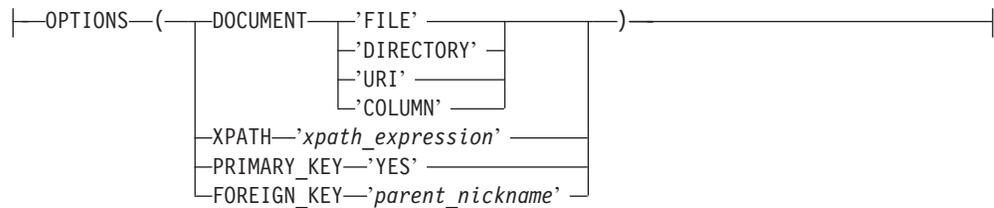
列構造:



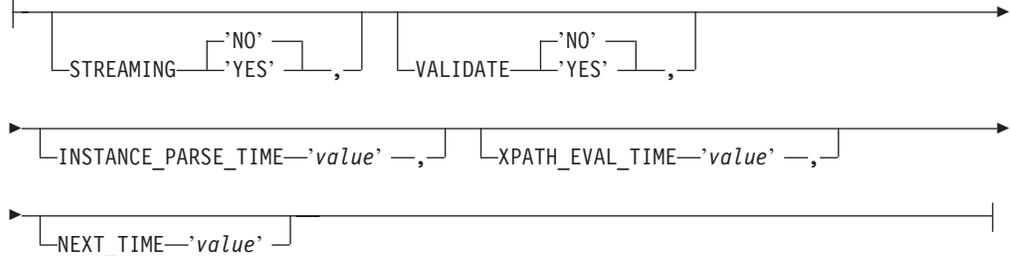
データ・タイプ・オプション:



ニックネーム列オプション:



ニックネーム・パラメーター:



制限: CLOB の長さは、XML ラッパーの場合 5 メガバイト (5MB) に制限されています。

ニックネーム・パラメーターおよびオプション:

FILE_PATH

XML 文書のファイル・パスを指定します。このニックネーム・オプションを指定する場合、DOCUMENT 列を指定しないでください。この FILE_PATH オプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) に対してのみ受け入れられます。

DIRECTORY_PATH

1 つ以上の XML ファイルを含むディレクトリーのパス名を指定します。このオプションは、複数の XML ソース・ファイルに単一のニックネームを作成するために使用します。XML ラッパーは、指定するディレクトリーにあるファイルのうち、.xml 拡張子を持つものだけを使用します。XML ラッパーは、このディレクトリーにある他のすべてのファイルを無視します。このニックネーム・オプションを指定する場合、DOCUMENT 列を指定しないでください。この DIRECTORY_PATH オプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) に対してのみ受け入れられます。

XPATH

個々のタプルを表す XML エレメントを識別する XPath 式を指定します。子ニックネームの XPATH ニックネーム・オプションは、その親の XPATH ニックネーム・オプションによって指定されるパスのコンテキストの中で評価されます。この XPath 式は、XPATH ニックネーム列オプションによって識別される列値を評価するためのコンテキストとして使用されます。

XPath 式にネーム・スペースの接頭部を指定しないでください。XML ラッパーはネーム・スペースをサポートしていません。

ニックネーム列オプション:

DOCUMENT

この列が DOCUMENT 列であることを指定します。DOCUMENT 列の値は、照会の実行時にニックネームに提供される XML ソース・データのタイプを示します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。DOCUMENT オプションでは、ニックネーム当たり 1 つの列しか指定できません。DOCUMENT オプションに関連付けられる列は VARCHAR データ・タイプでなければなりません。

FILE_PATH または DIRECTORY_PATH ニックネーム・オプションの代わりに DOCUMENT 列オプションを使用する場合、このニックネームに対応する文書が照会の実行時に提供されます。

DOCUMENT オプションに対する有効な値は以下のとおりです。

FILE ニックネーム列の値が XML 文書を含むファイルのパス名に結合されていることを指定します。このファイルからのデータが、照会の実行時に提供されます。

DIRECTORY

ニックネーム列の値が複数の XML データ・ファイルを含むディレクトリーのパス名に結合されていることを指定します。複数のファイルからの XML データが、照会の実行時に提供されます。このデータは指定されたディレクトリー・パスに存在する XML ファイルにあります。XML ラッパーは、指定するディレクトリーにあるファイルのうち、.xml 拡張子を持つものだけを使用します。XML ラッパーは、このディレクトリーにある他のすべてのファイルを無視します。

URI ニックネーム列の値が、URI が参照するリモート XML ファイルのパス名に結合されていることを指定します。この URI アドレスは、Web 上におけるこの XML ファイルのリモート・ロケーションを示します。

COLUMN

XML 文書がリレーショナル列に保管されることを指定します。

XPATH

XML 文書の中でこの列に対応するデータを含む XPath 式を指定します。CREATE NICKNAME ステートメントが XPATH ニックネーム・オプションから XPath 式を適用すると、XML ラッパーはその XPath 式を評価します。

大文字小文字の区別が不正確であるなどの、XPATH タグ参照が不正確に構成されている列名に対して照会を実行する場合、照会が必要なすべての列に対してその列にある NULL 値を戻します。

XPath 式にネーム・スペースの接頭部を指定しないでください。XML ラッパーはネーム・スペースをサポートしていません。

PRIMARY_KEY

このニックネームが親ニックネームであることを示します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。'YES' が唯一

の有効な値です。このオプションで指定した列に、ラッパーによって生成されたキーが保持されます。列の値は、SELECT 照会では検索できません。XPATH オプションをこの列に対して指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。

FOREIGN_KEY

このニックネームが子ニックネームであることを示し、対応する親ニックネームの名前を指定します。ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。このオプションで指定した列に、ラッパーによって生成されたキーが保持されます。列の値は、SELECT 照会では検索できません。XPATH オプションをこの列に対して指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。

親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを指定した CREATE NICKNAME ステートメントは失敗します。

FOREIGN_KEY 文節で参照されるニックネームが、対応する CREATE NICKNAME ステートメントのもとで引用符で囲むことによって、小文字または大文字小文字の混合として明示的に定義されていないかぎり、このニックネームを FOREIGN_KEY 文節で参照する場合、ニックネームを大文字で指定する必要があります。

ニックネーム・パラメーター:

STREAMING

XML ソース・ドキュメントをニックネームの XPath 式に一致するノードに対応する論理フラグメントに分けるかどうかを指定します。XML ラッパーは、XML ソース・データの構文解析と処理をフラグメントごとに行い、メモリーの使用合計量を減らします。どの XML ソース文書 (FILE、DIRECTORY、URI、または COLUMN) に対してもストリーミングを指定できます。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。デフォルトのストリーミング値は NO です。

VALIDATE パラメーターを YES に設定する場合、STREAMING パラメーターを YES に設定しないでください。両方のパラメーターを YES に設定すると、エラー・メッセージを受け取ります。

VALIDATE

XML データを抽出する前に XML ソース文書の妥当性検査を行うかどうかを指定します。このオプションを YES に設定すると、ニックネーム・オプションは、ソース文書の構造が XML スキーマまたは文書タイプ定義 (DTD) に準拠しているかどうかを検査します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。デフォルト値は NO です。

XML ラッパーが XML スキーマ・ファイルまたは DTD ファイル (.xsd または .dtd) を見つけることができない場合、XML ソース文書の妥当性検査は行われません。妥当性検査が行われなくても、DB2 はエラー・メッセ

ージを出しません。そのため、XML スキーマ・ファイルまたは DTD ファイルが XML ソース文書で指定されるロケーションに存在することを確認してください。

STREAMING パラメーターを YES に設定する場合、VALIDATE パラメーターを YES に設定しないでください。両方のパラメーターを YES に設定すると、エラー・メッセージを受け取ります。

INSTANCE_PARSE_TIME

XML ソース文書の 1 つの行にあるデータを構文解析する時間 (ミリ秒単位) を指定します。INSTANCE_PARSE_TIME、XPath_EVAL_TIME、および NEXT_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。指定する数は整数または 10 進数値です。デフォルト値は 7 ミリ秒です。

XPath_EVAL_TIME

ニックネームの XPath 式を評価して最初のエレメントを位置指定する時間 (ミリ秒単位) を指定します。XPath_EVAL_TIME、INSTANCE_PARSE_TIME、および NEXT_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションはルート・ニックネームおよび非ルート・ニックネーム用に使用できます。指定する数は整数または 10 進数値です。デフォルト値は 1 ミリ秒です。

NEXT_TIME

XPath 式からの後続のソース・エレメントを位置指定するために必要な時間 (ミリ秒単位) を指定します。NEXT_TIME、XPath_EVAL_TIME、および INSTANCE_PARSE_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションはルート・ニックネームおよび非ルート・ニックネーム用に使用できます。デフォルト値は 1 ミリ秒です。

使用上の注意:

DATE データ・タイプ・オプションを使用する場合、XML ソース文書にある日付は CCYY-MM-DD というフォーマットでなければなりません。たとえば、日付が 2002 年 11 月 17 日の場合、日付は XML ソース文書では 2002-11-17 と指定しなければなりません。日付が他のフォーマットであれば、エラー・メッセージを受け取ります。

STREAMING パラメーターと VALIDATE パラメーターの両方を YES に設定しないでください。XML ラッパーは XML ソース文書全体の妥当性検査を行い、ソース文書フラグメントの妥当性検査は行いません。両方のパラメーターを YES に設定すると、エラー・メッセージを受け取ります。

関連タスク:

- 474 ページの『XML データ・ソースのニックネームの登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『CREATE NICKNAME ステートメント』

- 474 ページの『CREATE NICKNAME ステートメント - XML ラッパーの例』

付録 A. フェデレーテッド情報を含むグローバル・カタログ表内のビュー

フェデレーテッド・データベース内のカタログ・ビューのほとんどは、他の DB2 for Linux, UNIX, and Windows データベース内のカタログ・ビューと同じです。いくつかの独自のビューとして、フェデレーテッド・システムに関係のある情報を含むビュー (SYSCAT.WRAPPERS ビューなど) があります。

DB2 for Linux, UNIX, and Windows バージョン 6 およびバージョン 7 の SQL リファレンス・マニュアルに記述されているように、DB2 バージョン 8 の SYSCAT ビューは現在、読み取り専用です。SYSCAT スキーマ内のビューに対して UPDATE または INSERT 操作を発行すると、失敗します。システム・カタログを更新する場合は SYSSTAT ビューを使用することをお勧めします。SYSCAT ビューを参照するアプリケーションは、SYSCAT ではなく更新可能な SYSSTAT ビューを参照するように変更してください。

次の表は、フェデレーテッド情報が入っている SYSCAT ビューをリストしています。これらのビューは読み取り専用です。

表 138. 通常、フェデレーテッド・システムで使用されるカタログ・ビュー

カタログ・ビュー	説明
SYSCAT.CHECKS	定義したチェック制約の情報が入っている。
SYSCAT.COLCHECKS	チェック制約によって参照される列が入っている。
SYSCAT.COLUMNS	ニックネームを作成したデータ・ソース・オブジェクト (表およびビュー) の列情報が入っている。
SYSCAT.COLOPTIONS	ニックネーム用にセットした列のオプション値の情報が入っている。
SYSCAT.CONSTDEP	定義した情報制約の従属関係が入っている。
SYSCAT.DATATYPES	ローカルの組み込み DB2 データ・タイプおよびユーザー定義 DB2 データ・タイプについてのデータ・タイプ情報が入っている。
SYSCAT.DBAUTH	個々のユーザーおよびグループが保持するデータベース権限が入っている。
SYSCAT.FUNCMAPOPTIONS	関数マッピング用にセットしたオプション値の情報が入っている。
SYSCAT.FUNCMAPPINGS	フェデレーテッド・データベースとデータ・ソース・オブジェクト間の関数マッピングが入っている。
SYSCAT.INDEXCOLUSE	索引に含める列が入っている。
SYSCAT.INDEXES	データ・ソース・オブジェクトの索引の指定が入っている。

表 138. 通常、フェデレーテッド・システムで使用されるカタログ・ビュー (続き)

カタログ・ビュー	説明
SYSCAT.KEYCOLUSE	ユニーク・キー、主キー、または外部キーの制約によって定義されるキーに加わる列が入っている。
SYSCAT.REFERENCES	自分で定義した参照制約に関する情報が入っている。
SYSCAT.ROUTINES	ローカル DB2 のユーザー定義関数または関数テンプレートが入っている。関数テンプレートは、データ・ソース関数と対応付けるために使用されます。
SYSCAT.REVTYPEMAPPINGS	このビューは使用されない。すべてのデータ・タイプ・マッピングは、SYSCAT.TYPEMAPPINGS ビューに記録されます。
SYSCAT.SERVEROPTIONS	サーバー定義でセットしたサーバーのオプション値の情報が入っている。
SYSCAT.SERVERS	データ・ソース・サーバー用に作成したサーバー定義が入っている。
SYSCAT.TABCONST	各行は、タイプ CHECK、UNIQUE、PRIMARY KEY、または FOREIGN KEY の表およびニックネームの制約を表す。
SYSCAT.TABLES	作成したそれぞれのローカル DB2 表、フェデレーテッド・ビュー、およびニックネームについての情報が入っている。
SYSCAT.TYPEMAPPINGS	フォワード・データ・タイプ・マッピングおよびリバース・データ・タイプ・マッピングが入っている。マッピングは、データ・ソースのデータ・タイプからローカル DB2 データ・タイプへの対応付け。これらのマッピングは、データ・ソース・オブジェクトでニックネームを作成するときに使用されます。
SYSCAT.USEROPTIONS	フェデレーテッド・データベースとデータ・ソース・サーバー間にユーザー・マッピングを作成した時にセットした、ユーザー権限情報が入っている。
SYSCAT.VIEWS	作成したローカル・フェデレーテッド・ビューについての情報が入っている。
SYSCAT.WRAPOPTIONS	ラッパーにセットしたオプション値についての情報が入っている。
SYSCAT.WRAPPERS	ラッパーを作成したそれぞれのデータ・ソースの、ラッパーおよびライブラリー・ファイルの名前が入っている。

次の表は、フェデレーテッド情報が入っている SYSSTAT ビューをリストしています。これらのビューは読み取り/書き込みビューであり、そこに入っている統計を更新することができます。

表 139. 更新可能なフェデレーテッド・グローバル・カタログ・ビュー

カタログ・ビュー	説明
SYSSTAT.COLUMNS	ニックネームを作成したデータ・ソース・オブジェクト (表およびビュー) の列についての統計情報が入っている。タイプされた表の継承された列については統計は記録されません。
SYSSTAT.INDEXES	データ・ソース・オブジェクトの索引の指定についての統計情報が入っている。
SYSSTAT.ROUTINES	ユーザー定義関数についての統計情報が入っている。組み込み関数は含まれない。タイプされた表の継承された列については統計は記録されません。
SYSSTAT.TABLES	基本表についての情報が入っている。このビューには、ビュー、シノニム、および別名の情報は含まれていません。タイプされた表の場合、表の階層のルート表だけがビューに含まれます。タイプされた表の継承された列については統計は記録されません。

付録 B. フェデレーテッド・システムのラッパー・オプション

ラッパー・オプションは、ラッパーを構成するため、またはフェデレーテッド・サーバーがどのようにラッパーを使用するかを定義するために使用されます。ラッパー・オプションは、ラッパーを作成または変更する際に設定できます。

すべてのリレーショナルおよび非リレーショナル・データ・ソースで、DB2_FENCED ラッパー・オプションを使用します。 ODBC データ・ソースは MODULE ラッパー・オプションを使用します。 Entrez データ・ソースは EMAIL ラッパー・オプションを使用します。

表 140. ラッパー・オプションとその設定値

オプション	有効な設定値	デフォルトの設定値
DB2_FENCED	ラッパーが fenced モードかトラステッド・モードのどちらで実行されるかを指定します。 Y ラッパーが fenced モードで実行されます。 N ラッパーがトラステッド・モードで実行されます。	リレーショナル・ラッパーの場合: N。 IBM 提供の非リレーショナル・ラッパーの場合: N。 サード・パーティー提供の非リレーショナル・ラッパーの場合: Y。
EMAIL	Entrez ラッパーを登録する際に E メール・アドレスを指定します。この E メール・アドレスはすべての照会に含められ、照会が多すぎて NCBI サーバーが過負荷になるなどの問題が生じる場合に、NCBI が連絡を取れるようにします。このオプションは必須です。	
MODULE	ODBC Driver Manager インプリメンテーションまたは SQL/CLI インプリメンテーションを含むライブラリーの絶対パスを指定する。UNIX フェデレーテッド・サーバー上の ODBC ラッパーの場合は必須。	Windows では、デフォルト値は odbc32.dll です。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『ニックネームを参照する照会の並列処理』

関連タスク:

- 「*IBM DB2 Information Integrator* ラッパー開発者向けガイド」の『トラステッドおよび fenced モードのプロセス環境』
- 「フェデレーテッド・システム・ガイド」の『ラッパーの変更』
- 68 ページの『データ・ソースのラッパーの登録』

付録 C. SQL ステートメントで有効なサーバーのタイプ

サーバー・タイプは、そのサーバー定義が表すデータ・ソースの種類を示します。サーバー・タイプは、ベンダー、目的、およびオペレーティング・システムに応じて変わります。サポートされる値は使用するラッパーによって異なります。

ほとんどのデータ・ソースで、CREATE SERVER ステートメントに有効なサーバー・タイプを指定しなければなりません。

BioRS ラッパー

BioRS データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要ありません。	BioRS

BLAST ラッパー

BLAST デーモンによってサポートされている BLAST データ・ソース。

サーバー・タイプ	データ・ソース
BLASTN	オリジナルの配列の領域と同種の領域を持つ配列を検出するために、ヌクレオチド配列をヌクレオチド配列データベースの内容と比較する BLAST 検索。
BLASTP	オリジナルの配列の領域と同種の領域を持つ配列を検出するために、アミノ酸配列をアミノ酸配列データベースの内容と比較する BLAST 検索。
BLASTX	オリジナルの配列の領域と同種の領域を持つ配列を検出するために、ヌクレオチド配列をアミノ酸配列データベースの内容と比較する BLAST 検索。
TBLASTN	オリジナルの配列の領域と同種の領域を持つ配列を検出するために、アミノ酸配列をヌクレオチド配列データベースの内容と比較する BLAST 検索。
TBLASTX	オリジナルの配列の領域と同種の領域を持つ配列を検出するために、ヌクレオチド配列をヌクレオチド配列データベースの内容と比較する BLAST 検索。

CTLIB ラッパー

CTLIB クライアント・ソフトウェアによりサポートされる Sybase データ・ソース。

サーバー・タイプ	データ・ソース
SYBASE	Sybase

Documentum ラッパー

Documentum Client API/Library によってサポートされている Documentum データ・ソース。

サーバー・タイプ	データ・ソース
DCTM	Documentum

DRDA ラッパー

DB2 ファミリーのデータ・ソース

表 141. DB2 for Linux, UNIX, and Windows

サーバー・タイプ	データ・ソース
DB2/UDB	IBM DB2 Universal Database
DB2/6000	IBM DB2 for AIX
DB2/AIX	IBM DB2 for AIX
DB2/HPUX	IBM DB2 for HP-UX
DB2/HP	IBM DB2 for HP-UX
DB2/NT	IBM DB2 for Windows NT
DB2/EEE	IBM DB2 Enterprise - Extended Edition
DB2/SUN	IBM DB2 for Solaris
DB2/PE	IBM DB2 for Personal Edition
DB2/2	IBM DB2 for OS/2
DB2/LINUX	IBM DB2 for Linux
DB2/PTX	IBM DB2 for NUMA-Q
DB2/SCO	IBM DB2 for SCO Unixware

表 142. DB2 for iSeries (および AS/400)

サーバー・タイプ	データ・ソース
DB2/400	IBM DB2 for iSeries および AS/400

表 143. DB2 for z/OS and OS/390

サーバー・タイプ	データ・ソース
DB2/ZOS	IBM DB2 for z/OS
DB2/390	IBM DB2 for OS/390
DB2/MVS	IBM DB2 for MVS

表 144. DB2 Server for VM and VSE

サーバー・タイプ	データ・ソース
DB2/VM	IBM DB2 for VM
DB2/VSE	IBM DB2 for VSE
SQL/DS	IBM SQL/DS

Entrez ラッパー

Entrez データ・ソース。

サーバー・タイプ	データ・ソース
NUCLEOTIDE	Entrez
PUBMED	Entrez

Excel ラッパー

Microsoft Excel 97、2000、および 2002 でサポートされている Excel データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要 ありません。	Microsoft Excel

Extended Search ラッパー

Extended Search クライアント・ライブラリーによってサポートされている Extended Search データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要 ありません。	IBM Lotus Extended Search

HMMER ラッパー

HMMER デーモンによってサポートされている HMMER データ・ソース。

サーバー・タイプ	データ・ソース
PFAM	HMMER

サーバー・タイプ	データ・ソース
SEARCH	HMMER

Informix ラッパー

Informix Client SDK ソフトウェアによりサポートされる Informix データ・ソース。

サーバー・タイプ	データ・ソース
INFORMIX	Informix

MSSQLODBC3 ラッパー

DataDirect Connect ODBC 3.6 ドライバー、または ODBC 3.0 以降のドライバーによってサポートされる Microsoft SQL Server データ・ソース。

サーバー・タイプ	データ・ソース
MSSQLSERVER	Microsoft SQL Server

NET8 ラッパー

Oracle NET8 クライアント・ソフトウェアによりサポートされる Oracle データ・ソース

サーバー・タイプ	データ・ソース
ORACLE	Oracle バージョン 8.0. またはそれ以上

ODBC ラッパー

ODBC 3.x ドライバーによりサポートされる ODBC データ・ソース。

サーバー・タイプ	データ・ソース
ODBC	ODBC

OLE DB ラッパー

Microsoft OLE DB 2.0 以上に準拠する OLE DB Provider。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要ありません。	任意の OLE DB Provider

表構造ファイル・ラッパー

表構造ファイル・データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要 ありません。	表構造ファイル

Teradata ラッパー

Teradata V2R3、V2R4、および V2R5 クライアント・ソフトウェアによってサポートされる Teradata データ・ソース。

サーバー・タイプ	データ・ソース
TERADATA	Teradata

Web サービス・ラッパー

Web サービス・データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要 ありません。	すべての Web サービス・データ・ソース。

WebSphere Business Integration ラッパー

WebSphere Business Integration ラッパーによってサポートされている、ビジネス・アプリケーション・データ・ソース。

サーバー・タイプ	データ・ソース
WBI	WebSphere Business Integration 2.2 または 2.3

XML ラッパー

XML データ・ソース。

サーバー・タイプ	データ・ソース
CREATE SERVER ステートメントでは必要 ありません。	XML

付録 D. フェデレーテッド・システムのサーバー・オプション

サーバー・オプションは、データ・ソース・サーバーを記述するために使用されます。サーバー・オプションは、データ保全性、ロケーション、セキュリティ、およびパフォーマンス情報を指定します。サーバー・オプションには、すべてのデータ・ソースで使用できるものもあれば、データ・ソースに固有のものもあります。

リレーショナル・データ・ソース用の共通のフェデレーテッド・サーバー・オプションは以下のとおりです。

- 互換性オプション。COLLATING_SEQUENCE、IGNORE_UDT
- データ保全性オプション。IUD_APP_SVPT_ENFORCE
- 日付および時間オプション。DATEFORMAT、 TIMEFORMAT、 TIMESTAMPFORMAT
- ロケーション・オプション。CONNECTSTRING、 DBNAME、 IFILE
- セキュリティー・オプション。FOLD_ID、 FOLD_PW、 INFORMIX_LOCK_MODE
- パフォーマンス・オプション。COMM_RATE、 CPU_RATIO、 DB2_MAXIMAL_PUSHDOWN、 IO_RATIO、 LOGIN_TIMEOUT、 PACKET_SIZE、 PLAN_HINTS、 PUSHDOWN、 TIMEOUT、 VARCHAR_NO_TRAILING_BLANKS

次の表は、各リレーショナル・データ・ソースに適用されるサーバー定義のサーバー・オプションをリストしています。

表 145. リレーショナル・データ・ソースのサーバー・オプション。

データ・ソース	CODEPAGE	COLLATING_SEQUENCE	COMM_RATE	CONNECTSTRING	CPU_RATIO	DATEFORMAT	DB2_MAXIMAL_PUSHDOWN	DBNAME	FOLD_ID	FOLD_PW	IFILE	INFORMIX_LOCK_MODE	IO_RATIO	IUD_APP_SVPT_ENFORCE	LOGIN_TIMEOUT	NODE	PACKET_SIZE	PASSWORD	PLAN_HINTS	PUSHDOWN	TIMEOUT	TIMEFORMAT	TIMESTAMPFORMAT	VARCHAR_NO_TRAILING_BLANKS	
DB2 UDB for iSeries		○	○		○		○	○	○	○			○	○				○		○					○
DB2 UDB for z/OS and OS/390		○	○		○		○	○	○	○			○	○				○		○					○
DB2 for VM and VSE		○	○		○		○	○	○	○			○	○				○		○					○

表 145. リレーショナル・データ・ソースのサーバー・オプション。(続き)

データ・ソース	CODEPAGE	COLLATING_SEQUENCE	COMM_RATE	CONNECTSTRING	CPU_RATIO	DATEFORMAT	DB2_MAXIMAL_PUSHDOWN	DBNAME	FOLD_ID	FOLD_PW	IFILE	INFORMIX_LOCK_MODE	IO_RATIO	IUD_APP_SVPT_ENFORCE	LOGIN_TIMEOUT	NODE	PACKET_SIZE	PASSWORD	PLAN_HINTS	PUSHDOWN	TIMEOUT	TIMEFORMAT	TIMESTAMPFORMAT	VARCHAR_NO_TRAILING_BLANKS
DB2 UDB for Linux, UNIX, and Windows		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>				<input type="checkbox"/>		<input type="checkbox"/>				<input type="checkbox"/>
Informix		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>				
Microsoft SQL Server	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>				
ODBC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
OLE DB		<input type="checkbox"/>		<input type="checkbox"/>																				
Oracle		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				<input type="checkbox"/>
Sybase		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>													
Teradata		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>						<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>								

次の表では、それぞれの非リレーショナル・データ・ソース (WebSphere Business Integration は除く) に適用可能な、サーバー定義のサーバー・オプションがリストされています。 631 ページの表 147 では、WebSphere Business Integration のサーバー定義のサーバー・オプションがリストされています。

表 146. 非リレーショナル・データ・ソースのサーバー・オプション

データ・ソース	CASE_SENSITIVE	CONTENT_DIR	DAEMON_PORT	ES_HOST	ES_PORT	ES_TRACING	ES_TRACELEVEL	ES_TRACEFILENAME	HMPFAM_OPTIONS	HMMSEARCH_OPTIONS	MAX_ROWS	NODE	OS_TYPE	PORT	PROCESSORS	PROXU_AUTHID	PROXY_PASSWORD	PROXY_SERVER_NAME	PROXY_SERVER_PORT	PROXY_TYPE	RDBMS_TYPE	SOCKET_TIMEOUT	TIMEOUT	TRANSACTIONS	USE_CLOB_SEQUENCE
BioRS	<input type="checkbox"/>											<input type="checkbox"/>		<input type="checkbox"/>									<input type="checkbox"/>		
BLAST			<input type="checkbox"/>									<input type="checkbox"/>	<input type="checkbox"/>												<input type="checkbox"/>
Documentum		<input type="checkbox"/>										<input type="checkbox"/>	<input type="checkbox"/>								<input type="checkbox"/>		<input type="checkbox"/>		
Entrez											<input type="checkbox"/>					<input type="checkbox"/>		<input type="checkbox"/>							
Excel																									

表 146. 非リレーショナル・データ・ソースのサーバー・オプション (続き)

データ・ソース	CASE_SENSITIVE	CONTENT_DIR	DAEMON_PORT	ES_HOST	ES_PORT	ES_TRACING	ES_TRACELEVEL	ES_TRACEFILENAME	HMPFAM_OPTIONS	HMMSEARCH_OPTIONS	MAX_ROWS	NODE	OS_TYPE	PORT	PROCESSORS	PROXU_AUTHID	PROXY_PASSWORD	PROXY_SERVER_NAME	PROXY_SERVER_PORT	PROXY_TYPE	RDBMS_TYPE	SOCKET_TIMEOUT	TIMEOUT	TRANSACTIONS	USE_CLOB_SEQUENCE
Extended Search				<input type="radio"/>																					
HMMER			<input type="radio"/>						<input type="radio"/>	<input type="radio"/>		<input type="radio"/>			<input type="radio"/>										<input type="radio"/>
表構造ファイル																									
Web サービス																									
XML																<input type="radio"/>		<input type="radio"/>							

次の表では、WebSphere Business Integration データ・ソースに適用可能な、サーバー一定義のサーバー・オプションがリストされています。

表 147. WebSphere Business Integration データ・ソースのサーバー・オプション

データ・ソース	APP_TYPE	FAULT_QUEUE	MQ_CONN_NAME	MQ_MANAGER	MQ_RESPONSE_TIMEOUT	MQ_SVRCONN_CHANNELNAME	REQUEST_QUEUE	RESPONSE_QUEUE
WebSphere Business Integration	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>					

次の表は、各サーバー・オプションを説明し、有効な設定値およびデフォルトの設定値をリストしています。

表 148. サーバー・オプションとその設定値

オプション	説明および有効な設定値	デフォルトの設定値
APP_TYPE	リモート・アプリケーションのタイプ。有効な値は 'PSOFT'、'SAP'、および 'SIEBEL' です。このオプションは必須です。	なし

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
CASE_SENSITIVE	<p>BioRS サーバーが大文字小文字を区別して名前を扱うかどうかを指定します。有効な値は Y または N です。</p> <p>'Y' BioRS サーバーが大文字小文字を区別して名前を扱います。</p> <p>'N' BioRS サーバーが大文字小文字を区別せずに名前を扱います。</p> <p>BioRS 製品では、BioRS サーバーに保管されているデータの 大文字小文字の区別を構成パラメーターで制御します。CASE_SENSITIVE オプションは、DB2 Information Integrator における BioRS システム構成パラメーターに相当します。BioRS システムと DB2 Information Integrator で、BioRS サーバーの大文字小文字の区別の構成設定を同期させなければなりません。BioRS と DB2 Information Integrator の間で大文字小文字の区別に関する構成設定を同期させておかないと、DB2 Information Integrator を介した BioRS データへのアクセスを試行した際にエラーが発生します。</p> <p>DB2 Information Integrator に新しい BioRS サーバーを作成した後で、CASE_SENSITIVE オプションを変更または削除することはできません。CASE_SENSITIVE オプションを変更する必要がある場合は、サーバー全体をドロップしてから再作成する必要があります。BioRS サーバーをドロップした場合、対応する BioRS ニックネームもすべて再作成しなければなりません。DB2 Information Integrator は、ドロップされたサーバーに対応するすべてのニックネームを自動的にドロップします。</p>	Y
CODEPAGE	<p>データ・ソースのクライアント構成のコード化文字セットに対応する、DB2 コード・ページ ID を指定します。クライアントのコード・ページとフェデレーテッド・データベースのコード・ページが一致しない場合、クライアントのコード・ページを指定しなければなりません。</p> <p>Unicode をサポートしているデータ・ソースの場合、データ・ソース・クライアントでのサポートされている Unicode エンコードに対応する、DB2 コード・ページ ID に CODEPAGE オプションを設定できます。</p>	<p>非 Unicode フェデレーテッド・データベースを使用する UNIX または Windows システムの場合: フェデレーテッド・データベース・コード・ページ。</p> <p>Unicode フェデレーテッド・データベースを使用する UNIX システムの場合: 1208</p> <p>Unicode フェデレーテッド・データベースを使用する Windows システムの場合: 1202</p>

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
COLLATING_SEQUENCE	<p>データ・ソースがフェデレーテッド・データベースと同じデフォルト照合シーケンスを使用するかどうかを、NLS コード・セットと国/地域別情報に基づいて指定します。</p> <p>'Y' データ・ソースは DB2 フェデレーテッド・データベースと同じ照合シーケンスを持っています。</p> <p>'N' データ・ソースは DB2 フェデレーテッド・データベースの照合シーケンスとは別の照合シーケンスを持っています。</p> <p>'I' データ・ソースは DB2 フェデレーテッド・データベースの照合シーケンスとは別の照合シーケンスを持ち、データ・ソースの照合シーケンスは大文字小文字の区別をしません (たとえば、'STEWART' と 'StewART' は等しいと見なされます)。</p>	'N'
COMM_RATE	<p>フェデレーテッド・サーバーとデータ・ソース・サーバー間の通信レートを指定します。秒当たりの MB 単位で表されます。</p> <p>有効な値は 0 より大きく、1×10^{23} より小さい値です。値は有効な REAL の表記で表せます。</p>	'2'
CONTENT_DIR	<p>GET_FILE、GET_FILE_DEL、GET_RENDITION、および GET_RENDITION_DEL 疑似列によって検索された内容ファイルを保管するための、ローカルにアクセス可能なルート・ディレクトリーの名前を指定します。これは、これらの疑似列を使用できるすべてのユーザーが書き込み可能でなければなりません。</p>	<p>UNIX システムの場合: '/tmp'</p> <p>Windows システムの場合: 'C:\%temp'</p>
CONNECTSTRING	<p>OLE DB Provider への接続に必要な初期化プロパティを指定します。</p>	なし
CPU_RATIO	<p>データ・ソースの CPU がフェデレーテッド・サーバーの CPU より、どれほど速いかまたは遅いかを示します。</p> <p>有効な値は 0 より大きく、1×10^{23} より小さい値です。値は有効な REAL の表記で表せます。</p> <p>設定値が 1 ということは、DB2 フェデレーテッドの CPU とデータ・ソースの CPU が、1:1 の比率で同じ速度になるということです。設定値が .5 ということは、DB2 フェデレーテッドの CPU の速度がデータ・ソースの CPU の速度の半分であるということです。設定値が 2 ということは、DB2 フェデレーテッドの CPU の速度がデータ・ソースの CPU の速度の 2 倍であるということです。</p>	'1.0'

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
DATEFORMAT	データ・ソースによって使用される日付形式。日付の数値形式を表現するには、'DD'、'MM'、および'YY' または 'YYYY' を使った形式を入力します。また、スペースやコンマなどの区切り文字を指定する必要もあります。たとえば、'2003-01-01' の日付形式を表現するには、'YYYY-MM-DD' を使用します。このフィールドは、NULL 可能です。	なし
DAEMON_PORT	デーモンが BLAST または HMMER ジョブ要求を listen するポート番号を指定します。ポート番号は、デーモン構成ファイルの DAEMON_PORT オプションに指定されているものと同じ番号でなければなりません。	BLAST の場合: '4007'。HMMER の場合: '4098'。
DB2_MAXIMAL_PUSHDOWN	照会オプティマイザーがアクセス・プランを選択するときに使用する 1 次基準を指定します。照会オプティマイザーは、コストを基に、またはリモート・データ・ソースによってできるだけ多くの照会処理を実行できるユーザー要件を基にアクセス・プランを選択できます。 'Y' 照会オプティマイザーは、照会操作を最も多くデータ・ソースにプッシュダウンするアクセス・プランを選択します。いくつかのアクセス・プランが同量のプッシュダウンを提供する場合、照会オプティマイザーは最もコストの低いプランを選択します。 フェデレーテッド・サーバー上のマテリアライズ照会表 (MQT) が一部あるいはすべての照会を処理できる場合は、マテリアライズ照会表を含むアクセス・プランが使用される可能性があります。フェデレーテッド・データベースは、カルテシアン積になる照会はプッシュダウンしません。 'N' 照会オプティマイザーは、コストを基にアクセス・プランを選択します。	'N'
DBNAME	フェデレーテッド・サーバーにアクセスさせるデータ・ソース・データベースの名前。DB2 データベースの場合、この値は最初のリモート DB2 データベース接続での特定のデータベースに対応しています。この特定のデータベースとは、CATALOG DATABASE コマンドまたは DB2 構成アシスタントを使用して、フェデレーテッド・サーバーでカタログされた、リモート DB2 データベースのデータベース別名です。Oracle インスタンスに含まれるのは 1 つのデータベースだけなので、Oracle データ・ソースには該当しません。	なし
ES_HOST	検索する Extended Search サーバーの完全修飾ホスト名または IP アドレスを指定します。このオプションは必須です。	なし

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
ES_PORT	この Extended Search が要求を listen するポート番号を指定します。このオプションは任意です。	'6001'
ES_TRACING	リモート Extended Search サーバーが作成するエラー・メッセージ、警告メッセージ、および通知メッセージに関して、トレースを有効にするかどうかを指定します。有効な値は次のとおりです。 'OFF' トレース・メッセージが記録されません。 'ON' トレース・メッセージが記録されます。このオプションは任意です。	'OFF'
ES_TRACELEVEL	トレースが有効になっている場合、このオプションはログ・ファイルに書き込まれるメッセージのタイプを指定します。以下のトレース・レベルを、個別に有効および無効にできます。 'C' 重大なエラー・メッセージ。 'N' 重大ではないエラー・メッセージ。 'W' 警告メッセージ。 'I' 通知メッセージ。 たとえば、次のようになります。 ES_TRACELEVEL 'W' ES_TRACELEVEL 'CN' このオプションは任意です。	'C'
ES_TRACEFILENAME	トレースが有効になっている場合、このオプションはメッセージが書き込まれるディレクトリーおよびファイルの名前を指定します。このオプションは任意です。	UNIX オペレーティング・システムの場合: \$INSTHOME/sqllib/log /ESWrapper.log Windows オペレーティング・システムの場合: %DB2TEMPDIR% ¥ESWrapper.log
FAULT_QUEUE	アダプターからラッパーへエラー・メッセージを送る障害キューの名前。この名前は、WebSphere MQ のキュー名に関する仕様に準拠していなければなりません。これは必須オプションです。	なし

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
FOLD_ID	<p>フェデレーテッド・サーバーが認証のためにデータ・ソース・サーバーに送信するユーザー ID に適用されます。有効な値は次のとおりです。</p>	なし
<p>(この表の最後にある注 1 および 4 を参照。)</p>	<p>'U' フェデレーテッド・サーバーは、ユーザー ID をデータ・ソースに送信する前に、大文字に変換します。これは、DB2 ファミリーおよび Oracle データ・ソースについては当然の選択です (この表の最後にある注 2 を参照。)</p>	
	<p>'N' フェデレーテッド・サーバーは、ユーザー ID をデータ・ソースに送信する前に、ユーザー ID に対して何の処理も行いません。(この表の最後の注 2 を参照。)</p>	
	<p>'L' フェデレーテッド・サーバーは、ユーザー ID をデータ・ソースに送信する前に、小文字に変換します。</p>	
	<p>これらの設定値のいずれも使用しない場合は、フェデレーテッド・サーバーはユーザー ID を大文字にしてデータ・ソースに送信しようとします。そのユーザー ID を正常に送信できない場合は、サーバーはユーザー ID を小文字で送信しようとします。</p>	
FOLD_PW	<p>フェデレーテッド・サーバーが認証のためにデータ・ソースに送信するパスワードに適用されます。有効な値は次のとおりです。</p>	なし
<p>(この表の最後にある注 1、3 および 4 を参照。)</p>	<p>'U' フェデレーテッド・サーバーは、パスワードをデータ・ソースに送信する前に、大文字に変換します。これは、DB2 ファミリーおよび Oracle データ・ソースについては当然の選択です。</p>	
	<p>'N' フェデレーテッド・サーバーは、パスワードをデータ・ソースに送信する前に、パスワードに対して何の処理も行いません。</p>	
	<p>'L' フェデレーテッド・サーバーは、パスワードをデータ・ソースに送信する前に、小文字に変換します。</p>	
	<p>これらの設定値のいずれも使用しない場合は、フェデレーテッド・サーバーはパスワードを大文字にしてデータ・ソースに送信しようとします。そのパスワードを正常に送信できない場合は、サーバーはパスワードを小文字で送信しようとします。</p>	

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
HMPFAM_OPTIONS	<p>オプションを列名にマップする参照表に対応する列名を持たない、hmpfam オプション (--null2、--pvm、--xnu など) を指定します。</p> <p>たとえば、以下のようになります。</p> <pre>HMPFAM_OPTIONS '--xnu --pvm'</pre> <p>この例では、デーモンは追加オプション --xnu、--pvm の他に、照会の WHERE 文節からオプションを指定して HMPFAM プログラムを実行します。</p>	
HMMSEARCH_OPTIONS	<p>ユーザーが hmmsearch コマンドに追加のコマンド行オプションを指定できるようにします。タイプ SEARCH でのみ有効です。詳細については、「HMMER User's Guide」を参照してください。</p>	なし
IFILE	<p>Sybase Open Client インターフェース・ファイルのパスと名前を指定します。Windows NT フェデレーテッド・サーバーでは、デフォルトは %DB2PATH%\%interfaces です。UNIX フェデレーテッド・サーバーでは、デフォルトのパスと名前の値は、\$DB2INSTANCE/sql/lib/interfaces です。</p>	なし
INFORMIX_LOCK_MODE	<p>Informix データ・ソースにロック・モードが設定されるように指定します。Informix ラッパーは、Informix データ・ソースとの接続を確立した直後に、'SET LOCK MODE' コマンドを発行します。有効な値は次のとおりです。</p> <p>'W' Informix ロック・モードを WAIT に設定します。ラッパーがロック表またはロック行へのアクセスを試みる場合、Informix はロックがリリースされるまで待ちます。</p> <p>'N' Informix ロック・モードを NOWAIT に設定します。ラッパーがロック表またはロック行へのアクセスを試みる場合、Informix はエラーを戻します。</p> <p>'n' Informix ロック・モードを WAIT <i>n</i> 秒に設定します。ラッパーがロック表またはロック行へのアクセスを試みるときに、指定した秒数内にリリースされない場合、Informix はエラーを戻します。</p>	'W'

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
IO_RATIO	<p>データ・ソース入出力システムがフェデレーテッド・サーバー入出力システムより、どれほど速いかまたは遅いかを表します。</p> <p>有効な値は 0 より大きく、1×10^{23} より小さい値です。値は有効な REAL の表記で表せます。</p> <p>設定値が 1 ということは、DB2 フェデレーテッド入出力とデータ・ソース入出力が、1:1 の比率で同じ速度になるということです。設定値が .5 ということは、DB2 フェデレーテッド入出力の速度がデータ・ソース入出力の速度の半分であるということです。設定値が 2 ということは、DB2 フェデレーテッド入出力の速度がデータ・ソース入出力の速度の 2 倍であるということです。</p>	'1.0'
IUD_APP_SVPT_ENFORCE	<p>DB2 フェデレーテッド・システムが、アプリケーション・セーブポイント・ステートメントの検出または作成を実施すべきかどうかを指定します。SET SERVER OPTION ステートメントを使用して設定すると、このサーバー・オプションは静的 SQL ステートメントに影響を与えません。</p> <p>'Y' フェデレーテッド・サーバーは、挿入、更新、または削除操作でエラーが発生する場合、およびデータ・ソースがアプリケーション・セーブポイント・ステートメントを実行しない場合、挿入、更新、または削除トランザクションをロールバックします。SQL エラー・コード SQL1476N が戻されます。</p> <p>'N' フェデレーテッド・サーバーは、エラーが発生した場合、トランザクションをロールバックしません。エラー・リカバリーは、アプリケーションが処理する必要があります。</p>	'Y'
LOGIN_TIMEOUT	<p>ログイン要求に対して、Sybase Open Client からの応答を DB2 フェデレーテッド・サーバーが待つ秒数を指定します。デフォルト値は TIMEOUT と同じです。</p>	'0'

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
MAX_ROWS	<p>Entrez ラッパーを使用する照会に関して、フェデレーテッド・サーバーが戻す行数を指定します。</p> <p>指定できるのは正数とゼロだけです。このオプションをゼロに設定すると、照会が NCBI Web サイトから検索できる行数が無制限になります。ただし、MAX_ROWS サーバー・オプションをゼロまたは大きい数に設定すると、照会のパフォーマンスに影響が及びます。</p> <p>MAX_ROWS サーバー・オプションは必須ではありません。</p>	<p>Microsoft Windows オペレーティング・システムの場合: 2000 行。</p> <p>UNIX ベースのオペレーティング・システムの場合: 5000 行。</p>
MQ_CONN_NAME	<p>Websphere MQ サーバーが実行されるコンピューターのホスト名またはネットワーク・アドレス。接続名は 9.30.76.151(1420) などになります。ここで、1420 はポート番号です。ポート番号が除かれている場合、デフォルト値である 1414 が使用されます。このオプションは任意です。これを省略すると、チャンネル定義を選択するために、MQSERVER 環境変数 (db2dj.ini ファイルで指定されている場合) が使用されます。MQSERVER が設定されていない場合、クライアントのチャンネル表が使用されます。</p>	<p>ラッパーは、チャンネル定義を選択するために、MQSERVER 環境変数 (db2dj.ini ファイルで指定されている場合) を使用します。MQSERVER 環境変数が設定されていない場合、ラッパーはクライアントのチャンネル表を使用します。</p>
MQ_MANAGER	<p>WebSphere MQ マネージャーの名前。任意の有効な WebSphere MQ マネージャー名です。このオプションは必須です。</p>	なし
MQ_RESPONSE_TIMEOUT	<p>ラッパーが応答キューから応答メッセージを待つ時間。値の単位はミリ秒です。特殊値として -1 を指定することにより、タイムアウト期間を設けないように指示できます。このオプションは任意です。</p>	10000
MQ_SVRCONN_CHANNELNAME	<p>ラッパーが接続を試行する、Websphere MQ Manager 上のサーバー接続チャンネルの名前。このパラメーターを指定できるのは、MQ_CONN_NAME サーバー・オプションが指定されている場合だけです。このオプションを省略すると、デフォルトのサーバー接続チャンネルである SYSTEM.DEF.SVRCONN が使用されます。</p>	SYSTEM.DEF.SVRCONN

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
NODE	リレーショナル・データ・ソースの場合: データ・ソースがインスタンスとして RDBMS に定義される際の名前。 Documentum の場合: Documentum Docbase の実際の名前を指定します。このオプションは必須です。 BLAST の場合: BLAST デーモン・プロセスが実行されるシステムのホスト名を指定します。このオプションは必須です。 HMMER の場合: HMMER デーモン・プロセスが実行されるサーバーのホスト名を指定します。このオプションは必須です。 BioRS の場合: BioRS 照会ツールが使用可能であるシステムのホスト名を指定します。このオプションは任意です。	BioRS の場合: <i>localhost</i>
OS_TYPE	Docbase サーバーのオペレーティング・システムを指定します。有効な値は AIX、SOLARIS、および WINDOWS です。このオプションは必須です。	なし
PACKET_SIZE	Sybase インターフェース・ファイルのケット・サイズをバイト単位で指定します。指定したケット・サイズをデータ・ソースがサポートしない場合、接続は失敗します。各レコードが非常に大きい場合 (たとえば、大きな表に行を挿入する場合など) にケット・サイズを増やすと、パフォーマンスが非常によくなります。バイト・サイズは数値です。	
PASSWORD	パスワードがデータ・ソースに送信されるかどうかを指定します。 'Y' パスワードがデータ・ソースに送信され、妥当性検査されます。 'N' パスワードがデータ・ソースに送信されず、妥当性検査が行われません。	'Y'

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
PLAN_HINTS	<p>プラン・ヒント を使用可能にするかどうかを指定します。プラン・ヒントはステートメントの一部であり、データ・ソース・オプティマイザーに対しての追加情報を提供します。特定の照会タイプについてこの情報を利用すれば、照会パフォーマンスを改善することができます。プラン・ヒントは、データ・ソース・オプティマイザーが索引を使用するかどうか、どの索引を使用するか、またはどの表結合シーケンスを使うかを判別するのに役立ちます。</p> <p>'Y' データ・ソースがプラン・ヒントをサポートしている場合は、プラン・ヒントを使用可能にします。</p> <p>'N' プラン・ヒントはデータ・ソースで使用可能になりません。</p> <p>このオプションは、Oracle および Sybase のデータ・ソースにのみ使用できます。</p>	'N'
PORT	BioRS サーバーへ接続するためにラッパーが使用するポートの番号を指定します。このオプションは任意です。	'5014'
PROCESSORS	HMMER プログラムが使用するプロセッサの番号を指定します。このオプションは、hmmfam コマンドの <code>--cpu</code> オプションに相当するものです。	なし
PROXY_AUTHID	PROXY_TYPE の値が 'SOCKS5' の場合に使用するユーザー名を指定します。PROXY_TYPE の値が 'SOCKS5' の場合、このフィールドはオプションです。使用するユーザー名については、ネットワーク管理者と連絡を取ってください。PROXY_TYPE が 'SOCKS5' ではない場合、このオプションは無効です。	なし
PROXY_PASSWORD	PROXY_TYPE の値が 'SOCKS5' の場合に使用するパスワードを指定します。PROXY_TYPE の値が 'SOCKS5' の場合、このフィールドはオプションです。使用するパスワードについては、ネットワーク管理者と連絡を取ってください。PROXY_TYPE が 'SOCKS5' ではない場合、このオプションは無効です。	なし
PROXY_SERVER_NAME	プロキシ・サーバー名または IP アドレスを指定します。PROXY_TYPE の値が 'HTTP'、'SOCKS4'、または 'SOCKS5' の場合、このフィールドは必須です。プロキシ・サーバー名または IP アドレスについては、ネットワーク管理者と連絡を取ってください。	なし

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
PROXY_SERVER_PORT	プロキシ・サーバーのポート番号を指定します。 PROXY_TYPE の値が 'HTTP'、'SOCKS4'、または 'SOCKS5' の場合、このフィールドは必須です。使用する必要があるプロキシ・サーバーのポート番号については、ネットワーク管理者と連絡を取ってください。	なし
PROXY_TYPE	ファイアウォールで保護された状態でインターネットにアクセスする際に使用する、プロキシ・タイプを指定します。有効なタイプは 'NONE'、'HTTP'、'SOCKS4'、または 'SOCKS5' です。デフォルト値は 'NONE' です。使用するプロキシのタイプについては、ネットワーク管理者と連絡を取ってください。	'NONE'
PUSHDOWN	'Y' DB2 UDB はデータ・ソースに操作を評価させることを考慮します。 'N' DB2 UDB は、列名を持つ SELECT のみを含むデータ・ソース SQL ステートメントを送信します。述部 (WHERE= など) 列、スカラー関数 (MAX や MIN など)、ソート (ORDER BY または GROUP BY など)、および結合は、データ・ソースに送信されるどの SQL にも含まれません。	'Y'
RDBMS_TYPE	Docbase で使用される RDBMS を指定します。有効な値は DB2、INFORMIX、ORACLE、SQLSERVER、または SYBASE です。このオプションは必須です。	なし
RESPONSE_QUEUE	アダプターからラッパーへ照会結果を送る応答キューの名前。この名前は、WebSphere MQ のキュー名に関する仕様に準拠していなければなりません。このオプションは必須です。	なし
REQUEST_QUEUE	ラッパーからアダプターへ照会結果を送る要求キューの名前。この名前は、WebSphere MQ のキュー名に関する仕様に準拠していなければなりません。このオプションは必須です。	なし
SOCKET_TIMEOUT	プロキシ・サーバーからの結果を DB2 フェデレーテッド・サーバーが待つ最大時間 (分単位) を指定します。有効な値は、ゼロ以上の任意の数です。デフォルトはゼロ '0' です。値ゼロは、待つ時間に制限を設けないことを意味します。	0
TIMEFORMAT	データ・ソースによって使用される時刻形式。 'hh12'、'hh24'、'mm'、'ss'、'AM'、または 'A.M' を使用した形式を入力します。たとえば、'16:00:00' という時刻形式を表現するには、'hh24:mm:ss' を使用します。'8:00:00 AM' という時刻形式を表現するには、'8:00:00 AM' を使用します。このフィールドは、NULL 可能です。	なし

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
TIMESTAMPFORMAT	データ・ソースによって使用されるタイム・スタンプ形式。形式は、日付、時刻、それに続いて 10 分の 1 秒の場合は 'n'、100 分の 1 秒の場合は 'nn'、1000 分の 1 秒 (ミリ秒) の場合は 'nnn' となります。最大でマイクロ秒の 'nnnnnn' まで可能です。たとえば、'2003-01-01-24:00:00.000000' のタイム・スタンプ形式を表現するには、'YYYY-MM-DD-hh24:mm:ss.nnnnnn' を使用します。このフィールドは、NULL 可能です。	なし
TIMEOUT	<p>Sybase の場合: 何らかの SQL ステートメントに対する Sybase Open Client からの応答を、DB2 フェデレーテッド・サーバーが待つ秒数を指定します。seconds の値は、DB2 Universal Database の整数範囲内にある、正の整数です。指定するタイムアウト値は、使用するラッパーにより異なります。Sybase ラッパーの TIMEOUT オプションのデフォルトの動作は 0 であり、これは DB2 UDB が無期限に応答を待つことを意味します。</p> <p>BioRS の場合: BioRS サーバーからの応答を BioRS ラッパーが待つ時間 (分単位) を指定します。デフォルト値は 10 です。このオプションは任意です。</p>	'0'。BioRS の場合は '10'。
TRANSACTIONS	<p>サーバー・トランザクション・モードを指定します。有効な値は以下のとおりです。</p> <p>'NONE' 使用可能なトランザクションはありません。</p> <p>'QUERY' Dctm_Query メソッドに限り、トランザクションが使用可能です。</p> <p>'ALL' Dctm_Query メソッドに関して、トランザクションが使用可能です。このリリースでは、ALL と QUERY の機能は同じです。</p>	'QUERY'
USE_CLOB_SEQUENCE	このオプションは、BlastSeq または HmmQSeq 列に関してフェデレーテッド・サーバーが使用するデータ・タイプを指定します。可能な値は 'Y' または 'N' です。CREATE NICKNAME または ALTER NICKNAME ステートメントを使用して、BlastSeq または HmmQSeq 列のデフォルト・データ・タイプをオーバーライドできます。	'Y'

表 148. サーバー・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
VARCHAR_NO_ TRAILING_BLANKS	このオプションは、比較の際にデータ・ソースに可変長の文字データ・タイプがあり、その可変長データが末尾ブランクで埋められていない場合に適用されます。	N (影響を受けるデータ・ソースに対して)。
	ある種のデータ・ソース (Oracle など) は、ブランク埋め込み文字比較セマンティクスを持たず、これは DB2 for Linux, UNIX, and Windows の比較セマンティクスと同じ結果を戻します。このオプションは、指定されたサーバーからアクセスされるデータ・ソース・オブジェクト内の、すべての VARCHAR および VARCHAR2 列にこれを適用したい場合に設定してください。これにはビューも含まれます。	
	Y これらの VARCHAR 列に末尾ブランクがないか、またはこのデータ・ソースに、フェデレーテッド・サーバー上のセマンティクスと類似した、ブランク埋め込み文字比較セマンティクスが含まれています。	
	フェデレーテッド・サーバーは、文字比較演算をデータ・ソースにプッシュして、処理させます。	
	N これらの VARCHAR 列に末尾ブランクがあり、データ・ソースに、フェデレーテッド・サーバーとは異なるブランク埋め込み文字比較セマンティクスが含まれていません。	
	相当するセマンティクスを補うことができない場合に、フェデレーテッド・サーバーは文字比較演算を処理します。たとえば、述部の再作成などです。	

この表に関する注

- このフィールドは、認証に指定される値に関係なく適用されます。
- DB2 UDB はユーザー ID を大文字で保管するので、値 'N' と 'U' は論理的に互いに同等です。
- パスワードの設定が 'N' の場合は、FOLD_PW を設定しても効果はありません。パスワードが送信されないため、大文字小文字の区別は意味をなしません。
- これらのいずれのオプションについても、NULL 値を設定することは避けてください。NULL 値を設定すると、ユーザー ID とパスワードを解決するために DB2 UDB は複数回試行することになるので、便利のように思えますが、パフォーマンスが悪化する可能性があります (DB2 UDB がユーザー ID とパスワードを 4 回送信した後で、ようやくデータ・ソース認証に成功するということもあり得ます)。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『プッシュダウンの可能性に影響を与えるサーバー特性』
- 「フェデレーテッド・システム・ガイド」の『グローバルな最適化に影響を与えるサーバー特性』

関連タスク:

- 69 ページの『データ・ソースのサーバー定義の登録』

関連資料:

- 「SQL リファレンス 第 2 巻」の『DROP ステートメント』
- 「SQL リファレンス 第 2 巻」の『ALTER SERVER ステートメント』
- 「SQL リファレンス 第 2 巻」の『CREATE SERVER ステートメント』

付録 E. フェデレーテッド・システムのユーザー・マッピング・オプション

これらのオプションは、すべてのリレーショナル・データ・ソースで有効です。非リレーショナル・データ・ソースの場合、REMOTE_AUTHID および REMOTE_PASSWORD オプションは、BioRS、Documentum、Extended Search、および Web サービスのデータ・ソースで有効です。GUEST オプションは BioRS データ・ソースで有効です。

以下のオプションは、CREATE USER MAPPING および ALTER USER MAPPING ステートメントで使用されます。

表 149. ユーザー・マッピング・オプションおよびその設定値

オプション	有効な設定値	デフォルトの設定値
ACCOUNTING	DRDA の場合: DRDA 会計情報ストリングの指定に使用します。有効な設定値は、長さが 255 以下の任意のストリングです。このオプションは、会計情報を渡す必要がある場合のみ必要です。詳細は、「DB2 Connect ユーザーズ・ガイド」を参照してください。	なし
GUEST	ラッパーがゲスト・アクセス・モードを使用して BioRS サーバーにアクセスするかどうかを指定します。 Y ラッパーが BioRS サーバーへのゲスト・アクセス・モードを使用します。 N ラッパーが BioRS サーバーへのゲスト・アクセス・モードを使用しません。 値を Y に設定すると、このオプションは REMOTE_AUTHID オプションおよび REMOTE_PASSWORD オプションと相互に排他的になります。	N
REMOTE_AUTHID	データ・ソースで使用される許可 ID を表します。有効な設定値は、長さが 255 以下の任意のストリングです。	DB2 Universal Database への接続に使用する許可 ID。
REMOTE_DOMAIN	Documentum の場合: Documentum データ・ソースに接続するユーザーの認証に使用される、Windows NT ドメインを表します。有効な設定値は、任意の有効な Windows NT ドメイン・ネームです。	Documentum データベースのデフォルトの認証ドメイン。

表 149. ユーザー・マッピング・オプションおよびその設定値 (続き)

オプション	有効な設定値	デフォルトの設定値
REMOTE_PASSWORD	<p>データ・ソースで使用される許可パスワードを表します。有効な設定値は、長さが 32 以下の任意のストリングです。</p> <p>以下の条件が満たされている場合、このオプションを設定する必要はありません。</p> <ul style="list-style-type: none"> データベース・マネージャーの構成パラメーター AUTHENTICATON が SERVER に設定されている。 DB2 データベースへの接続時に認証 ID およびパスワードを指定した。 <p>サーバーからパスワードが要求され、このオプションを設定しない場合、前述の条件が両方とも満たされていることを確認してください。そうでなければ、接続は失敗します。</p>	<p>有効な設定の列にリストされている両方の条件が満たされている場合、DB2 Universal Database への接続に使用するパスワード。</p>

関連概念:

- 「DB2 Connect ユーザーズ・ガイド」の『DB2 Connect と DRDA』
- 「DB2 Connect ユーザーズ・ガイド」の『DRDA とデータ・アクセス』

関連タスク:

- 70 ページの『データ・ソースのユーザー・マッピングの登録』

付録 F. フェデレーテッド・システムのニックネーム・オプション

表 150 および表 151 では、各データ・ソースのニックネーム・オプションがリストされています。 650 ページの表 152 では、各ニックネーム・オプションについて説明し、有効な設定とデフォルト設定がリストされています。

表 150. 使用可能なニックネーム・オプション - A から P

データ・ソース	ALL_VERSIONS	APPLICATIONID	BUSOBJ_NAME	CATEGORY	COLUMN_DELIMITER	DATASOURCE	DIRECTORY_PATH	FILE_PATH	FOLDERS	HMMTYPE	INSTANCE_PARSE_TIME	IS_REG_TABLE	KEY_COLUMN	MAXHIT	NAMESPACES	NEXT_TIME	PARENT	PROCESSORS
BioRS																		
BLAST						○												○
Documentum	○								○			○						
Entrez																	○	
Excel								○										
Extended Search		○		○		○								○				
HMMER						○				○								
表構造ファイル					○			○					○					
Web サービス															○			
WebSphere Business Integration			○												○			
XML							○	○			○					○		

表 151 では、各データ・ソースのニックネーム・オプションの R から X までがリストされています。

表 151. 使用可能なニックネーム・オプション - R から X

データ・ソース	RANGE	REMOTE_OBJECT	SOAPACTION	SORTED	SORTFIELD	SORTORDER	STREAMING	TEMPLATE	TOTALMAXHIT	TIMEOUT	URL	VALIDATE	VALIDATE_DATA_FILE	VERTICAL_TABLE	XPATH	XPATH_EVAL_TIME
BioRS		○														
BLAST										○						
Documentum		○														
Entrez		○														

表 151. 使用可能なニックネーム・オプション - R から X (続き)

データ・ソース	RANGE	REMOTE_OBJECT	SOAPACTION	SORTED	SORTFIELD	SORTORDER	STREAMING	TEMPLATE	TOTALMAXHIT	TIMEOUT	URL	VALIDATE	VALIDATE_DATA_FILE	VERTICAL_TABLE	XPATH	XPATH_EVAL_TIME
Excel	○															
Extended Search					○	○			○	○				○		
HMMER										○						
表構造ファイル				○									○			
Web サービス			○				○	○			○				○	
WebSphere Business Integration								○							○	
XML							○					○			○	○

表 152 では、各ニックネーム・オプションについて説明し、有効な設定とデフォルト設定がリストされています。

表 152. ニックネーム・オプションとその設定値

オプション	説明および有効な設定値	デフォルトの設定値
ALL_VERSIONS	すべてのオブジェクト・バージョンを検索するかどうか指定します。有効な値は y、Y、n、および N です。デフォルト値の N は、現行オブジェクト・バージョンのみが照会処理に組み込まれることを意味します。このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。	N
APPLICATIONID	検索する Extended Search アプリケーションの名前を指定します。この名前は、Extended Search 構成データベースに存在していなければなりません。このオプションは必須です。	
BUSOBJ_NAME	ビジネス・オブジェクトを表す XML スキーマ定義ファイル (.xsd) の名前。たとえば、sap_bapi_customer_get_detail2 などです。このオプションは、親ニックネームで指定しなければなりません。	
CATEGORY	検索する 1 つ以上の Extended Search カテゴリを指定します。このオプションを省略する場合、少なくとも 1 つのデータ・ソース名を指定しなければなりません。複数のカテゴリを指定するには、カテゴリ名をセミコロンで区切ります。	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
COLUMN_DELIMITER	<p>単一引用符で囲まれている、表構造ファイルの列を区切るのに使用される区切り文字。列区切り文字には、複数の文字分の長さが可能です。列区切り文字が定義されていない場合、デフォルトの区切り文字はコンマです。単一引用符は区切り文字としては使用できません。列区切り文字は、ファイル全体で一貫性がなければなりません。 NULL 値は 2 つの区切り文字が続くことによって表されます。 NULL フィールドが行の末尾にある場合、区切り文字に行終止符を続けることによって表されます。列区切り文字は、列の有効なデータとして存在することはできません。</p>	<p>デフォルトの区切り文字はコンマです。</p>
DATASOURCE	<p>Extended Search の場合: 検索する 1 つ以上の Extended Search データ・ソースを指定します。このオプションを省略する場合、少なくとも 1 つのカテゴリ名を指定しなければなりません。複数のデータ・ソースを指定するには、データ・ソース名をセミコロンで区切ります。</p> <p>BLAST の場合: BLAST 検索を実行するデータ・ソースの名前。ここで使用されるのと同じストリングが、BLAST デーモンの構成ファイル内に存在していなければなりません。このオプションは必須です。</p> <p>HMMER (タイプ PFAM) の場合: HMMPFAM によって検索される HMM プロファイル・データベース。ここで使用されるのと同じストリングが、HMMER デーモンの構成ファイル内に存在していなければなりません。このオプションは必須です。</p> <p>HMMER (タイプ SEARCH) の場合: HMMSEARCH によって検索されるシーケンス・ファイルの名前。ここで使用されるのと同じストリングが、HMMER デーモンの構成ファイル内に存在していなければなりません。このオプションは必須です。</p>	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
<p>DIRECTORY_PATH</p>	<p>1 つ以上の XML ファイルを含むディレクトリーのパス名を指定します。このオプションは、複数の XML ソース・ファイルに単一のニックネームを作成するために使用します。XML ラッパーは、指定するディレクトリー内にある、.xml 拡張子を持つファイルだけを使用します。XML ラッパーは、このディレクトリーにある他のすべてのファイルを無視します。このニックネーム・オプションを指定する場合、DOCUMENT 列を指定しないでください。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) に対してのみ受け入れられます。</p>	
<p>FILE_PATH</p>	<p>Microsoft Excel の場合: アクセスする Excel スプレッドシートの完全修飾ディレクトリー・パスおよびファイル名を指定します。このオプションは必須です。</p> <p>表構造ファイルの場合: アクセスされる表構造ファイルへの完全修飾パス (単一引用符で囲む)。データ・ファイルは標準ファイルまたはシンボリック・リンクでなければならず、パイプや別の非標準ファイル・タイプにすることはできません。FILE_PATH または DOCUMENT ニックネーム列オプションのいずれかを指定しなければなりません。FILE_PATH ニックネーム・オプションが指定されている場合には、DOCUMENT ニックネーム列オプションは指定できません。</p> <p>XML の場合: XML 文書のファイル・パスを指定します。このニックネーム・オプションを指定する場合、DOCUMENT 列を指定しないでください。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) に対してのみ受け入れられます。</p>	
<p>FOLDERS</p>	<p>1 つ以上の論理的に結合され、構文的に正確な Documentum FOLDER 述部を含むストリングを指定します。FOLDER 述部を指定すると、このニックネームによって表される文書のセットを、指定したフォルダー内の文書に制限します。</p> <p>このオプションを指定する場合、FOLDERS オプションの値全体を単一引用符で囲み、ストリング内では単一引用符の代わりに二重引用符を使用します。</p> <p>このオプションは、IS_REG_TABLE = 'Y' の場合は無効になります。</p>	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
HMMTYPE	オプション: モデルと遺伝子配列の両方で使用するアルファベット。値は NUCLEIC か PROTEIN のいずれかで、大文字小文字の区別はありません。	PROTEIN
INSTANCE_PARSE_TIME	XML ソース文書の 1 つの行にあるデータを構文解析する時間 (ミリ秒単位) を指定します。 INSTANCE_PARSE_TIME、 XPATH_EVAL_TIME、および NEXT_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。指定する数は整数または 10 進数値です。	7
IS_REG_TABLE	REMOTE_OBJECT オプションによって指定されたオブジェクトが、Documentum の登録済みの表かどうかを指定します。有効な値は 'y'、'Y'、'n'、および 'N' です。 ALTER NICKNAME ステートメントでこのオプションを変更して、ニックネームを Documentum オブジェクトから登録済みの表に (またはその逆に) 変更することはできません。そうする代わりに、ニックネームをドロップし、再作成する必要があります。	N
KEY_COLUMN	単一引用符で囲まれている、ファイルをソートするキーとなるファイル内の列の名前。このオプションは、ソート・ファイルでのみ使用してください。DOCUMENT ニックネーム列オプションで指定された列は、キー列として指定してはなりません。 単一系列のキーのみがサポートされます。複数列のキーは許可されていません。値は、CREATE NICKNAME ステートメントに定義されている列の名前にしなければなりません。列は必ず昇順でソートされます。NOT NULL オプションをニックネーム・ステートメントの中のその定義に追加することにより、キー列が NULL 可能ではないということを指定しなければなりません。 このオプションには大文字小文字の区別がありません。ただし 列が二重引用符付きで定義されていない限り、DB2 UDB によって列名が大文字に変更されます。	保管されているニックネームに関して値が指定されていない場合、値は、ニックネーム・ファイル内の最初の列の名前になります。
MAXHIT	検索されている各ソースから戻される結果の最大数を指定する INTEGER です。	50

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
NAMESPACES	<p>各列の XPATH および TEMPLATE オプションで使用される、ネーム・スペース接頭部に関連付けられているネーム・スペース。構文は以下のとおりです。</p> <pre>NAMESPACES 'prefix1="actual_namespace1", prefix2="actual_namespace2" '</pre> <p>各ネーム・スペースをコンマで区切ってください。たとえば、以下のようになります。</p> <pre>NAMESPACES ' c="http://www.myweb.com/cust", i="http://www.myweb.com/cust/id", n="http://www.myweb.com/cust/name"'</pre>	
NEXT_TIME	<p>XPath 式からの後続のソース・エレメントを位置指定するために必要な時間 (ミリ秒単位) を指定します。NEXT_TIME、XPath_EVAL_TIME、および INSTANCE_PARSE_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションはルート・ニックネームおよび非ルート・ニックネーム用に使用できます。</p>	1
PARENT	<p>REMOTE_OBJECT オプションによって親の名前変更が行われた、子ニックネームに対してのみ指定されます。PARENT オプションは、DB2 スキーマ内で複数のニックネーム・ファミリーが定義されている場合に、子を親に関連付けます。この名前は大文字小文字が区別されます。</p>	
PROCESSORS	<p>BLAST 照会の評価時に使用されるプロセッサの数を指定します。このオプションは、blastall -a オプションに対応しています。</p>	1
RANGE	<p>データ・ソースで使用されるセルの範囲を指定します。</p>	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
REMOTE_OBJECT	<p>BioRS の場合: ニックネームに関連する BioRS データ・バンクの名前を指定します。この名前によって、ニックネームのスキーマと BioRS データ・バンクが決まります。この名前によって、そのニックネームと他のニックネームの関係も指定されます。この名前で大文字小文字を区別するかどうかは、BioRS サーバーで大文字小文字を区別するかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。</p> <p>ALTER NICKNAME ステートメントを使用して、この名前の変更または削除を行うことはできません。このオプションで使用される BioRS データ・バンクの名前を変更した場合、ニックネーム全体を削除してから、再びそれを再作成しなければなりません。</p> <p>Documentum の場合: ニックネームに関連した Documentum オブジェクト・タイプの名前を指定します。名前は任意の Documentum オブジェクト・タイプまたは登録済みの表にできます。登録済みの表の名前には、接頭部として表の所有者の名前を付けなければなりません。登録済みの表が Docbase 所有者に所属する場合、値 dm_dbo を所有者名に使用できます。このオプションは必須です。新しいオブジェクトの構造がオリジナルのオプションの構造に類似していない場合、ALTER NICKNAME ステートメントを使用して REMOTE_OBJECT オプションの値を変更するとエラーになります。</p> <p>Entrez の場合: ニックネームに関連した Entrez オブジェクト・タイプの名前を指定します。この名前によって、ニックネームのスキーマと NCBI データベース、および他のニックネームとの関係が決定されます。この名前では大文字小文字が区別されません。</p>	
SOAPACTION	<p>Web サービス記述言語 (WSDL) 形式からの URI SOAPACTION 属性。このオプションは、ルート・ニックネームごとに必要です。このオプションを、非ルート・ニックネームで使用することはできません。</p>	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
SORTED	<p>データ・ソース・ファイルがソートされるか、ソートされないかを指定します。このオプションでは、Y、y、n、または N を指定できます。</p> <p>ソートされるデータ・ソースは、LC_COLLATE 各国語サポート・カテゴリーの設定によって定義されている現在のロケールの照合順序に従って、昇順でソートされなければなりません。</p> <p>データ・ソースがソートされるように指定する場合、VALIDATE_DATA_FILE オプションを Y に設定してください。</p>	N
SORTFIELD	<p>検索結果がソートされなければならないフィールドの名前を指定します。デフォルト値の DOC_RANK は、Extended Search が検索文書の関連性を判別するのに使用するフィールドです。別のフィールド名を指定する場合、検索するソースにその名前が存在していなければなりません。</p>	DOC_RANK
SORTORDER	<p>検索結果の戻りのソート順序 (昇順 (A) か降順 (D) のいずれか) を指定します。</p>	A
STREAMING	<p>XML ソース文書を処理するために、論理フラグメントに分ける必要があるかどうかを指定します。このフラグメントは、ニックネームの XPath 式に一致するノードに対応しています。XML ラッパーは、XML ソース・データの構文解析と処理をフラグメントごとに行います。このタイプの構文解析では、メモリーの使用量が最小限度で済みます。このオプションは、ルート・ニックネームでしか指定できません。</p> <p>どの XML ソース文書 (FILE、DIRECTORY、URI、または COLUMN) に対してもストリーミングを指定できます。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。</p> <p>有効な値は次のとおりです。</p> <p>Y XML 文書は構文解析されます。</p> <p>N XML 文書は構文解析されません。</p> <p>VALIDATE パラメーターを YES に設定する場合、STREAMING パラメーターを YES に設定しないでください。両方のパラメーターを YES に設定すると、エラー・メッセージを受け取ります。</p>	N

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
TEMPLATE	<p>WebSphere Business Integration の場合: XML 入力文書を構成するために使用する、ニックネーム・テンプレートのフラグメント。フラグメントは、指定されたテンプレートの構文に準拠していなければなりません。</p> <p>Web サービスの場合: SOAP 要求を構成するために使用する、ニックネーム・テンプレートのフラグメント。フラグメントは、指定されたテンプレートの構文に準拠していなければなりません。</p>	
TOTALMAXHIT	<p>検索されているすべてのソースから戻される結果の最大数を指定する INTEGER です。ラッパーはそれらの結果を 1 つの結果セットに結合します。</p>	50
TIMEOUT	<p>Extended Search の場合: 要求のタイムアウトになるまでサーバーからの応答を待つ秒数を指定する INTEGER。</p> <p>BLAST および HMMER の場合: ラッパーがデーモンからの結果を待つ最大時間 (分単位)。</p>	<p>Extended Search の場合: 30。</p> <p>BLAST および HMMER の場合: 60。</p>
URL	<p>Web サービス・エンドポイントの URL。このオプションは、ルート・ニックネームごとに必要です。このオプションを、非ルート・ニックネームで使用することはできません。サポートされているプロトコルは HTTP および HTTPS です。</p>	

表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
VALIDATE	<p>XML データを抽出する前に XML ソース文書の妥当性検査を行うかどうかを指定します。このオプションを YES に設定すると、ニックネーム・オプションは、ソース文書の構造が XML スキーマまたは文書タイプ定義 (DTD) に準拠しているかどうかを検査します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。</p> <p>XML ラッパーが XML スキーマ・ファイルまたは DTD ファイル (.xsd または .dtd) を見つけることができない場合、XML ソース文書の妥当性検査は行われません。妥当性検査が行われない場合、DB2 UDB はエラー・メッセージを出します。そのため、XML スキーマ・ファイルまたは DTD ファイルが XML ソース文書で指定されるロケーションに存在することを確認してください。</p> <p>STREAMING パラメーターを YES に設定する場合、VALIDATE パラメーターを YES に設定しないでください。両方のパラメーターを YES に設定すると、エラー・メッセージを受け取ります。</p>	NO
VALIDATE_DATA_FILE	<p>ソートされるファイルの場合、このオプションは、キー列が昇順でソートされていることをラッパーがチェックするかどうか、および NULL キーをチェックするかどうかを指定します。このオプションで有効な値は Y または N だけです。この検査は、登録時に一度だけ行われます。</p> <p>DOCUMENT ニックネーム列オプションがファイル・パスに使用されている場合には、このオプションは使用できません。</p>	N
VERTICAL_TABLE	<p>検索結果の表示フォーマットを指定します。YES を指定すると、Extended Search はユーザー定義列のほかに、戻せるものとして構成されているすべてのフィールドを戻します。ラッパーは列名の垂直リストとして結果をニックネーム表に保管します。</p>	NO

| 表 152. ニックネーム・オプションとその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
XPATH	<p>個々のタブルを表すエレメントを識別する XPath 式を指定します。子ニックネームの XPATH ニックネーム・オプションは、その親の XPATH ニックネーム・オプションによって指定されるパスのコンテキストの中で評価されます。この XPath 式は、XPATH ニックネーム列オプションによって識別される列値を評価するためのコンテキストとして使用されます。</p> <p>XML の場合: XPATH 式にネーム・スペース接頭部を指定しないでください。XML ラッパーはネーム・スペースをサポートしていません。</p>	
XPATH_EVAL_TIME	<p>ニックネームの XPath 式を評価して最初のエレメントを位置指定する時間 (ミリ秒単位) を指定します。</p> <p>XPATH_EVAL_TIME、INSTANCE_PARSE_TIME、および NEXT_TIME オプションを変更して、大規模または複雑な XML ソース構造の照会を最適化できます。このオプションはルート・ニックネームおよび非ルート・ニックネーム用に使用できます。指定する数は整数または 10 進数値です。</p>	1

付録 G. フェデレーテッド・システムのニックネーム列オプション

列情報は、ニックネーム列オプションと呼ばれるパラメーターを使用して、CREATE NICKNAME または ALTER NICKNAME ステートメントで指定することができます。

以下の表では、各データ・ソースのニックネーム列オプションがリストされています。

表 153. 使用可能なニックネーム列オプション

データ・ソース	ALL_VALUES	DEFAULT	DELIMITER	DOCUMENT	ESCAPE_INPUT	FOREIGN_KEY	INDEX	IS_REPEATING	NUMERIC_STRING	PRIMARY_KEY	REMOTE_NAME	SOAPACTIONCOLUMN	TEMPLATE	URLCOLUMN	VARCHAR_NO_TRAILING_BLANKS	XPATH
BLAST		<input type="radio"/>	<input type="radio"/>				<input type="radio"/>									
DB2 Universal Database for iSeries									<input type="radio"/>							
DB2 Universal Database for z/OS and OS/390									<input type="radio"/>							
DB2 Universal Database for VM and VSE									<input type="radio"/>							
DB2 Universal Database for Linux, UNIX, and Windows									<input type="radio"/>							
Documentum	<input type="radio"/>		<input type="radio"/>					<input type="radio"/>			<input type="radio"/>					
Informix									<input type="radio"/>							
Microsoft SQL Server									<input type="radio"/>							
ODBC									<input type="radio"/>							
OLE DB									<input type="radio"/>							
Oracle									<input type="radio"/>						<input type="radio"/>	
Sybase									<input type="radio"/>							
表構造ファイル				<input type="radio"/>												

表 153. 使用可能なニックネーム列オプション (続き)

データ・ソース	ALL_VALUES	DEFAULT	DELIMITER	DOCUMENT	ESCAPE_INPUT	FOREIGN_KEY	INDEX	IS_REPEATING	NUMERIC_STRING	PRIMARY_KEY	REMOTE_NAME	SOAPACTIONCOLUMN	TEMPLATE	URLCOLUMN	VARCHAR_NO_TRAILING_BLANKS	XPATH
Teradata									○							
WebSphere Business Integration					○	○				○			○			○
Web サービス					○	○				○		○	○	○		○
XML				○		○				○						○

表 154. 列オプションおよびその設定値

オプション	説明および有効な設定値	デフォルトの設定値
ALL_VALUES	<p>繰り返し属性のすべての値を、指定された区切り文字で分離して戻すことを指定します。このオプションが指定されていない場合、または N の場合、繰り返し属性の最後の値だけが戻されます。</p> <p>ALL_VALUES オプションを指定できるのは、IS_REPEATING が 'Y' (IS_REG_TABLE = 'Y' の場合は無効) である VARCHAR 列だけです。</p>	
DEFAULT	<p>以下の入力固定列の新しいデフォルト値を指定します。</p> <ul style="list-style-type: none"> • E_value • QueryStrands • GapAlign • NMisMatchPenalty • NMatchReward • Matrix • FilterSequence • NumberOfAlignments • GapCost • ExtendedGapCost • WordSize • ThresholdEx <p>この新しい値は、事前設定されているデフォルト値をオーバーライドします。新しいデフォルト値は、その列の指示されている値と同じタイプでなければなりません。</p>	

表 154. 列オプションおよびその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
DELIMITER	<p>Documentum の場合: 繰り返し属性の複数の値を連結する場合に使用される、区切り文字ストリングを指定します。区切り文字は 1 文字以上です。このオプションが有効なのは、IS_REPEATING オプションが Y に設定されている、データ・タイプ VARCHAR のオブジェクトの属性に限ります。</p> <p>BLAST の場合: このオプションが示される列の、定義情報の終点を判別するために使用される区切り文字。このオプションの値で複数の文字が示されている場合、それらの文字のいずれかが最初に出現した位置が、このフィールドの情報の最後であることを表します。デフォルトは行の終わりです。最後に指定した列に、定義行の残りを入れる場合を除いて、このオプションは必須です。</p>	<p>Documentum の場合: デフォルトの区切り文字はコンマです。</p> <p>BLAST: デフォルトの区切り文字は行の終わりです。</p>

表 154. 列オプションおよびその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
DOCUMENT	<p>表構造ファイルの場合: 表構造ファイルの種類を指定します。このラッパーは、このオプションで値 <code>FILE</code> のみをサポートしています。DOCUMENT オプションでは、ニックネーム当たり 1 つの列しか指定できません。DOCUMENT オプションに関連付けられる列は、データ・タイプ <code>VARCHAR</code> または <code>CHAR</code> の列でなければなりません。</p> <p>FILE_PATH ニックネーム・オプションではなく、DOCUMENT ニックネーム列オプションを使用するということは、照会の実行中に、このニックネームに対応するファイルが提供されることを意味します。DOCUMENT オプションに <code>FILE</code> 値が指定されている場合、照会の実行時に提供される値は、このニックネームのニックネーム定義に一致するスキーマを持つファイルの絶対パスです。</p> <p>XML の場合: この列が DOCUMENT 列であることを指定します。DOCUMENT 列の値は、照会の実行時にニックネームに提供される XML ソース・データのタイプを示します。このオプションは、ルート・ニックネーム (XML 文書の最上位にあるエレメントを識別するニックネーム) の列に対してのみ受け入れられます。DOCUMENT オプションでは、ニックネーム当たり 1 つの列しか指定できません。DOCUMENT オプションに関連付けられる列は <code>VARCHAR</code> データ・タイプでなければなりません。</p> <p>FILE_PATH または DIRECTORY_PATH ニックネーム・オプションの代わりに DOCUMENT 列オプションを使用する場合、このニックネームに対応する文書が照会の実行時に提供されます。</p> <p>DOCUMENT オプションに対する有効な値は、以下のとおりです。</p> <p>FILE ニックネーム列の値がファイルのパス名に結合されていることを指定します。このファイルからのデータは、照会の実行時に提供されます。</p> <p>DIRECTORY ニックネーム列の値が複数の XML データ・ファイルを含むディレクトリーのパス名に結合されていることを指定します。複数のファイルからの XML データが、照会の実行時に提供されます。このデータは、指定されたディレクトリー・パス内の XML ファイルにあります。XML ラッパーは、指定するディレクトリー内にある、.xml 拡張子を持つファイルだけを使用します。XML ラッパーは、このディレクトリーにある他のすべてのファイルを無視します。</p> <p>URI ニックネーム列の値が、URI が参照するリモート XML ファイルのパス名に結合されていることを指定します。この URI アドレスは、Web 上におけるこの XML ファイルのリモート・ロケーションを示します。</p> <p>COLUMN XML 文書がリレーショナル列に保管されることを指定します。</p>	
ELEMENT_NAME	<p>BioRS エレメント名を指定します。この名前で大文字小文字を区別するかどうかは、BioRS サーバーで大文字小文字を区別するかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。BioRS エレメント名の指定が必須なのは、エレメント名が列名と異なる場合だけです。</p>	

表 154. 列オプションおよびその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
ESCAPE_INPUT	<p>XML 特殊文字が XML 入力値で置換されるかどうかを指定します。このオプションは、繰り返しエレメントを含む XML フラグメントなど、XML フラグメントを入力として含めるために使用します。TEMPLATE 列オプションは、ESCAPE_INPUT 列オプションを使用する列で定義しなければなりません。列データ・タイプは、VARCHAR または CHAR でなければなりません。</p> <p>有効な値は次のとおりです。</p> <p>Y XML 入力に特殊文字が含まれている場合、それらは置換され、入力文字を示すために XML で使用される対応する文字に変えられます。</p> <p>N 入力文字が、完全にそのままの形で保持されます。</p>	Y
FOREIGN_KEY	<p>このニックネームが子ニックネームであることを示し、対応する親ニックネームの名前を指定します。ニックネームには、FOREIGN_KEY 列オプションを 1 つしか指定できません。このオプションの値は、大文字小文字を区別します。このオプションで指定した表に、ラッパーによって生成されたキーが保持されます。XPath オプションをこの列に対して指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。</p> <p>親ニックネームに別のスキーマ名が付けられている場合、FOREIGN_KEY オプションを指定した CREATE NICKNAME ステートメントは失敗します。</p> <p>FOREIGN_KEY 文節で参照されるニックネームを FOREIGN_KEY 文節で参照する場合、ニックネームを大文字で指定する必要があります。ただしそのニックネームを、対応する CREATE NICKNAME ステートメント内で引用符で囲むことによって、小文字または大文字小文字の混合として明示的に定義してある場合は例外です。</p> <p>このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>	
INDEX	<p>定義行の列のグループの中で、このオプションが現れる列の順序。このオプションは必須です。</p>	
IS_INDEXED	<p>対応する列が索引付けされているか (述部で列が参照可能か) を示します。有効な値は 'Y' と 'N' です。値 'Y' を指定できるのは、対応するエレメントが BioRS サーバーによって索引付けされている列に対してだけです。</p>	<p>ニックネームの作成時に、このオプションには値 'Y' が設定されて、BioRS によって索引付けられたエレメントに対応するすべての列に追加されます。</p>
IS_REPEATING	<p>列が複数値かどうかを示します。有効な値は Y および N です。</p> <p>以下の場合は、最後の値のみが戻されます。</p> <ul style="list-style-type: none"> 非 VARCHAR 繰り返し属性 ALL_VALUES 'N' が指定されているときの VARCHAR 列 <p>この制限に対処するため、繰り返し属性列に二重の定義を作成できます。</p>	N

表 154. 列オプションおよびその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
NUMERIC_STRING	<p>列に数字のストリングが含まれているかどうかを指定します。</p> <p>Y この列には、'0'、'1'、'2'、.... '9' という、数字のストリングが含まれます。ブランクは含みません。この列に、数値ストリングと末尾ブランクだけが含まれる場合は、Y を指定しないでください。</p> <p>列に対して NUMERIC_STRING を Y に設定すると、列データをソートする場合に支障となり得るブランクがこの列には含まれないことを、最適マイザーに知らせることになります。データ・ソースの照合シーケンスが、フェデレーテッド・サーバーが使用する照合シーケンスとは異なる場合に、このオプションを使用してください。このオプションを使用する列は、照合シーケンスが異なるためにリモートでの評価から除かれるということはありません。</p> <p>N この列は数値ストリング列ではないか、またはブランクを含む数値ストリング列であるかのどちらかです。</p>	N
PRIMARY_KEY	<p>このニックネームが親ニックネームであることを示します。列データ・タイプは VARCHAR(16) でなければなりません。ニックネームには、PRIMARY_KEY 列オプションを 1 つしか指定できません。YES が唯一の有効な値です。このオプションで指定した列に、ラッパーによって生成されたキーが保持されます。XPath オプションをこの列に対して指定してはなりません。列は、親のニックネームと子のニックネームを結合するためだけに使用できます。</p> <p>このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>	
REFERENCED_OBJECT	<p>このオプションは、BioRS データ・タイプが Reference である列にのみ有効です。このオプションは、現在の列が参照する BioRS データ・バンクの名前を指定します。この名前で大文字小文字を区別するかどうかは、BioRS サーバーで大文字小文字を区別するかどうか、および CASE_SENSITIVE サーバー・オプションの値によって決まります。</p>	
REMOTE_NAME	<p>対応する Documentum 属性または列の名前を指定します。このオプションは、リモート属性または列名をローカルの DB2 UDB 列名にマップします。</p>	DB2 UDB 列名。
SOAPACTIONCOLUMN	<p>Web サービス記述言語 (WSDL) 形式からの URI SOAPACTION 属性を動的に指定する列。このオプションは、ルート・ニックネームでしか指定できません。</p> <p>このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>	
TEMPLATE	<p>XML 入力文書を構成するために使用する、列テンプレートのフラグメント。フラグメントは、指定されたテンプレートの構文に準拠していなければなりません。</p>	
URLCOLUMN	<p>照会の実行時に、Web サービスのエンドポイントの URL を動的に指定する列。このオプションは、ルート・ニックネームでしか指定できません。</p> <p>このオプションが列に設定されている場合、その列に他のオプションを設定することはできません。</p>	

表 154. 列オプションおよびその設定値 (続き)

オプション	説明および有効な設定値	デフォルトの設定値
VARCHAR_NO_ TRAILING_BLANKS	このオプションは、比較の際にデータ・ソースに可変長の文字データ・タイプがあり、その可変長データが末尾ブランクで埋められていない場合に適用されます。	N (影響を受けるデータ・ソースに対して)
	ある種のデータ・ソース (Oracle など) は、ブランク埋め込み文字比較セマンティクスを持たず、これは DB2 UDB for Linux, UNIX, and Windows の比較セマンティクスと同じ結果を戻します。このオプションは、データ・ソース・オブジェクト内の特定の VARCHAR または VARCHAR2 列だけに適用したい場合に設定してください。	
	Y これらの VARCHAR 列に末尾ブランクがないか、またはこのデータ・ソースに、フェデレーテッド・サーバー上のセマンティクスと類似した、ブランク埋め込み文字比較セマンティクスが含まれています。	
	フェデレーテッド・サーバーは、文字比較演算を処理のためにデータ・ソースに送信します。	
	N これらの VARCHAR 列に末尾ブランクがあり、データ・ソースに、フェデレーテッド・サーバーとは異なるブランク埋め込み文字比較セマンティクスが含まれていません。	
	相当するセマンティクスを補うことができない場合に、フェデレーテッド・サーバーは文字比較演算を処理しません。たとえば、述部の再作成などです。	
XPATH	XML 文書の中でこの列に対応するデータを含む XPath 式を指定します。CREATE NICKNAME ステートメントがこの XPATH ニックネーム・オプションから XPath 式を適用した後で、ラッパーはその XPath 式を評価します。	

関連概念:

- 「フェデレーテッド・システム・ガイド」の『プッシュダウン分析』

関連タスク:

- 「フェデレーテッド・システム・ガイド」の『グローバルな最適化』

付録 H. デフォルトの順方向データ・タイプ・マッピング

データ・ソースのデータ・タイプとフェデレーテッド・データベースのデータ・タイプ間のマッピングには、順方向タイプのマッピングと逆方向タイプのマッピングの 2 種類があります。順方向タイプ・マッピング では、マッピングはリモートのタイプから対応するローカル・タイプへのマッピングです。

デフォルトのタイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新しいタイプ・マッピングを作成することもできます。

これらのマッピングは、特に記述のないかぎり、サポート対象のすべてのバージョンで有効です。

データ・ソースから DB2 for Linux, UNIX, and Windows へのデフォルトの順方向データ・タイプ・マッピングのすべてについて、DB2 フェデレーテッド・スキーマは SYSIBM です。

次の表は、DB2 for Linux, UNIX, and Windows のデータ・タイプとデータ・ソースのデータ・タイプ間のデフォルトの順方向マッピングを示しています。

DB2 for z/OS and OS/390 データ・ソース

表 155. DB2 for z/OS and OS/390 のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-

表 155. DB2 for z/OS and OS/390 のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
1 FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
1 ROWID	-	-	-	-	Y	-	VARCHAR	40	-	Y
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N
VARGGRAPHIC	1	16336	-	-	-	-	VARGGRAPHIC	-	0	N

DB2 for iSeries データ・ソース

表 156. DB2 for iSeries のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	255	32672	-	-	-	-	VARCHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CHAR	255	32672	-	-	Y	-	VARCHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-

表 156. DB2 for iSeries のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
I FLOAT	4	-	-	-	-	-	REAL	-	-	-
I FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
GRAPHIC	128	16336	-	-	-	-	VARGRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
NUMERIC	-	-	-	-	-	-	DECIMAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARG	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 Server for VM and VSE データ・ソース

表 157. DB2 Server for VM and VSE のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-

表 157. DB2 Server for VM and VSE のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
CHAR	1	254	-	-	-	-	CHAR	-	0	N
CHAR	1	254	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBAHW	-	-	-	-	-	-	SMALLINT	-	0	-
DBAINT	-	-	-	-	-	-	INTEGER	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
FLOAT	4	-	-	-	-	-	REAL	-	-	-
FLOAT	8	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	1	127	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	0	N
VARCHAR	1	32672	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPH	1	16336	-	-	-	-	VARGRAPHIC	-	0	N

DB2 for Linux、UNIX、および Windows のデータ・ソース

表 158. DB2 for Linux, UNIX, and Windows のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BIGINT	-	-	-	-	-	-	BIGINT	-	0	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHAR	-	-	-	-	-	-	CHAR	-	0	N
CHAR	-	-	-	-	Y	-	CHAR	-	0	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	0	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	-	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	-	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	0	N
INTEGER	-	-	-	-	-	-	INTEGER	-	0	-
LONGVAR	-	-	-	-	N	-	CLOB	-	-	-
LONGVAR	-	-	-	-	Y	-	BLOB	-	-	-
LONGVARG	-	-	-	-	-	-	DBCLOB	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	0	-
TIME	-	-	-	-	-	-	TIME	-	0	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
TIMESTMP	-	-	-	-	-	-	TIMESTAMP	-	0	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	0	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	0	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	0	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	0	N

Informix データ・ソース

表 159. Informix のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	2147483647	-	-
BOOLEAN	-	-	-	-	-	-	CHARACTER	1	-	-
BYTE	-	-	-	-	-	-	BLOB	2147483647	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CLOB	-	-	-	-	-	-	CLOB	2147483647	-	-
DATE	-	-	-	-	-	-	DATE	4	-	-
DATETIME	0	4	0	4	-	-	DATE	4	-	-
DATETIME	6	10	6	10	-	-	TIME	3	-	-
DATETIME	0	4	6	15	-	-	TIMESTAMP	10	-	-
DATETIME	6	10	11	15	-	-	TIMESTAMP	10	-	-
DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
DECIMAL	32	130	-	-	-	-	DOUBLE	8	-	-
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	-	-	-

表 159. Informix のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
-----------------	------------------	------------------	--------------------	--------------------	-----------------	-----------------------	--------------------	------------------	-----------------	--------------------

注:

Informix DATETIME データ・タイプでは、DB2 UNIX and Windows フェデレーテッド・サーバーは、Informix 高水準修飾子を REMOTE_LENGTH として、Informix 低水準修飾子を REMOTE_SCALE として使用します。

Informix 修飾子は、Informix クライアント SDK datatype.h ファイルに定義されている "TU_" 定数です。定数は次のとおりです。

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)	

Microsoft SQL Server データ・ソース

表 160. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
bigint ⁴	-	-	-	-	-	-	BIGINT	-	-	-
binary	1	254	-	-	-	-	CHARACTER	-	-	Y
binary	255	8000	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	2	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	8000	-	-	-	-	VARCHAR	-	-	N
datetime	-	-	-	-	-	-	TIMESTAMP	10	-	-
datetimen	-	-	-	-	-	-	TIMESTAMP	10	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-

表 160. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
DUMMY65 ¹	1	38	-84	127	-	-	DOUBLE	-	-	-
DUMMY2000 ³	1	38	-84	127	-	-	DOUBLE	-	-	-
float	-	8	-	-	-	-	DOUBLE	8	-	-
floatn	-	8	-	-	-	-	DOUBLE	8	-	-
float	-	4	-	-	-	-	REAL	4	-	-
floatn	-	4	-	-	-	-	REAL	4	-	-
image	-	-	-	-	-	-	BLOB	2147483647	-	Y
int	-	-	-	-	-	-	INTEGER	4	-	-
intn	-	-	-	-	-	-	INTEGER	4	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	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	-	-
numericn	32	38	0	38	-	-	DOUBLE	-	-	-
numericn	1	31	0	31	-	-	DECIMAL	-	-	-
ntext ²	-	-	-	-	-	-	CLOB	2147483647	-	Y
nvarchar	1	4000	-	-	-	-	VARCHAR	-	-	N
real	-	-	-	-	-	-	REAL	4	-	-
smallint	-	-	-	-	-	-	SMALLINT	2	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP	10	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
smallmoneyn	-	-	-	-	-	-	DECIMAL	10	4	-
SQL_BIGINT	-	-	-	-	-	-	DECIMAL	-	-	-
SQL_BIGINT ⁴	-	-	-	-	-	-	BIGINT	-	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N

表 160. Microsoft SQL Server のデフォルトの順方向データ・タイプ・マッピング (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
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_DECIMAL	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_GUID ²	1	4000	-	-	Y	-	VARCHAR	16	-	Y
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_REAL	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	8000	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	8000	-	-	-	-	VARCHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	N
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	2	-	-
uniqueidentifier ²	1	4000	-	-	Y	-	VARCHAR	16	-	Y
varbinary	1	8000	-	-	-	-	VARCHAR	-	-	Y
varchar	1	8000	-	-	-	-	VARCHAR	-	-	N

注:

- | 1. このタイプ・マッピングは Microsoft SQL Server バージョン 6.5 でのみ有効です。
- | 2. このタイプ・マッピングは Microsoft SQL Server バージョン 7 およびバージョン 2000 でのみ有効です。
- | 3. このタイプ・マッピングは Windows 2000 オペレーティング・システムでのみ有効です。
- | 4. このタイプ・マッピングは Microsoft SQL Server バージョン 2000 でのみ有効です。

ODBC データ・ソース

表 161. ODBC のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
SQL_BIGINT	-	-	-	-	-	-	BIGINT	8	-	-
SQL_BINARY	1	254	-	-	-	-	CHARACTER	-	-	Y
SQL_BINARY	255	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_BIT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_CHAR	1	254	-	-	-	-	CHAR	-	-	N
SQL_CHAR	255	32672	-	-	-	-	VARCHAR	-	-	N
SQL_DECIMAL	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_DECIMAL	32	38	0	38	-	-	DOUBLE	8	-	-
SQL_DOUBLE	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_FLOAT	-	-	-	-	-	-	DOUBLE	8	-	-
SQL_INTEGER	-	-	-	-	-	-	INTEGER	4	-	-
SQL_LONGVARCHAR	-	-	-	-	-	-	CLOB	2147483647	-	N
SQL_LONGVARBINARY	-	-	-	-	-	-	BLOB	-	-	Y
SQL_NUMERIC	1	31	0	31	-	-	DECIMAL	-	-	-
SQL_NUMERIC	32	32	0	31	-	-	DOUBLE	8	-	-
SQL_REAL	-	-	-	-	-	-	REAL	4	-	-
SQL_SMALLINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_TYPE_DATE	-	-	-	-	-	-	DATE	4	-	-
SQL_TYPE_TIME	-	-	-	-	-	-	TIME	3	-	-
SQL_TYPE_TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	10	-	-
SQL_TINYINT	-	-	-	-	-	-	SMALLINT	2	-	-
SQL_VARBINARY	1	32672	-	-	-	-	VARCHAR	-	-	Y
SQL_VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	N
SQL_WCHAR	1	127	-	-	-	-	CHAR	-	-	N
SQL_WCHAR	128	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WVARCHAR	1	16336	-	-	-	-	VARCHAR	-	-	N
SQL_WLONGVARCHAR	-	1073741823	-	-	-	-	CLOB	2147483647	-	N

Oracle NET8 データ・ソース

表 162. Oracle NET8 のデフォルトの順方向データ・タイプ・マッピング

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BLOB	0	0	0	0	-	¥0	BLOB	2147483647	0	Y
CHAR	1	254	0	0	-	¥0	CHAR	0	0	N
CHAR	255	2000	0	0	-	¥0	VARCHAR	0	0	N
CLOB	0	0	0	0	-	¥0	CLOB	2147483647	0	N
DATE	0	0	0	0	-	¥0	TIMESTAMP	0	0	N
FLOAT	1	126	0	0	-	¥0	DOUBLE	0	0	N
LONG	0	0	0	0	-	¥0	CLOB	2147483647	0	N
LONG RAW	0	0	0	0	-	¥0	BLOB	2147483647	0	Y
MLSLABEL	0	0	0	0	-	¥0	VARCHAR	255	0	N
NUMBER	1	38	-84	127	-	¥0	DOUBLE	0	0	N
NUMBER	1	31	0	31	-	>=	DECIMAL	0	0	N
NUMBER	1	4	0	0	-	¥0	SMALLINT	0	0	N
NUMBER	5	9	0	0	-	¥0	INTEGER	0	0	N
NUMBER	-	10	0	0	-	¥0	DECIMAL	0	0	N
RAW	1	2000	0	0	-	¥0	VARCHAR	0	0	Y
ROWID	0	0	0	NULL	-	¥0	CHAR	18	0	N
TIMESTAMP ¹	-	-	-	-	-	-	TIMESTAMP	10	-	-
VARCHAR2	1	4000	0	0	-	¥0	VARCHAR	0	0	N

注:

1. このタイプ・マッピングは、Oracle 9i (またはそれ以降) のクライアントおよびサーバー構成でのみ有効です。

Sybase データ・ソース

表 163. Sybase CTLIB のデフォルトの順方向データ・タイプ・マッピング

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
binary	1	254	-	-	-	-	CHAR	-	-	Y
binary	255	16384	-	-	-	-	VARCHAR	-	-	Y
bit	-	-	-	-	-	-	SMALLINT	-	-	-
char	1	254	-	-	-	-	CHAR	-	-	N
char	255	16384	-	-	-	-	VARCHAR	-	-	N
char null (varchar を参照)										
datetime	-	-	-	-	-	-	TIMESTAMP	-	-	-
datetimn	-	-	-	-	-	-	TIMESTAMP	-	-	-
decimal	1	31	0	31	-	-	DECIMAL	-	-	-
decimal	32	38	0	38	-	-	DOUBLE	-	-	-
decimaln	1	31	0	31	-	-	DECIMAL	-	-	-
decimaln	32	38	0	38	-	-	DOUBLE	-	-	-
float	-	4	-	-	-	-	REAL	-	-	-
float	-	8	-	-	-	-	DOUBLE	-	-	-
floatn	-	4	-	-	-	-	REAL	-	-	-
floatn	-	8	-	-	-	-	DOUBLE	-	-	-
image	-	-	-	-	-	-	BLOB	-	-	-
int	-	-	-	-	-	-	INTEGER	-	-	-
intn	-	-	-	-	-	-	INTEGER	-	-	-
money	-	-	-	-	-	-	DECIMAL	19	4	-
moneyn	-	-	-	-	-	-	DECIMAL	19	4	-
nchar	1	254	-	-	-	-	CHAR	-	-	N
nchar	255	16384	-	-	-	-	VARCHAR	-	-	N
nchar null (nvarchar を参照)										
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	16384	-	-	-	-	VARCHAR	-	-	N

表 163. Sybase CTLIB のデフォルトの順方向データ・タイプ・マッピング (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
real	-	-	-	-	-	-	REAL	-	-	-
smalldatetime	-	-	-	-	-	-	TIMESTAMP	-	-	-
smallint	-	-	-	-	-	-	SMALLINT	-	-	-
smallmoney	-	-	-	-	-	-	DECIMAL	10	4	-
sysname	1	254	-	-	-	-	CHAR	-	-	N
text	-	-	-	-	-	-	CLOB	-	-	-
timestamp	-	-	-	-	-	-	VARCHAR	8	-	Y
tinyint	-	-	-	-	-	-	SMALLINT	-	-	-
unichar ¹	1	254	-	-	-	-	CHAR	-	-	N
unichar ¹	255	16384	-	-	-	-	VARCHAR	-	-	N
unichar null (univarchar を参照)										
univarchar ¹	1	16384	-	-	-	-	VARCHAR	-	-	N
varbinary	1	16384	-	-	-	-	VARCHAR	-	-	Y
varchar	1	16384	-	-	-	-	VARCHAR	-	-	N

注:

1. 非 Unicode フェデレーテッド・データベースの場合に有効。

Teradata データ・ソース

表 164. Teradata のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BYTE	1	254	-	-	-	-	CHAR	-	-	Y
BYTE	255	32672	-	-	-	-	VARCHAR	-	-	Y

表 164. Teradata のデフォルトの順方向データ・タイプ・マッピング (表示されていない列があります) (続き)

REMOTE_TYPENAME	REMOTE_LOWER_LEN	REMOTE_UPPER_LEN	REMOTE_LOWER_SCALE	REMOTE_UPPER_SCALE	REMOTE_BIT_DATA	REMOTE_DATA_OPERATORS	FEDERATED_TYPENAME	FEDERATED_LENGTH	FEDERATED_SCALE	FEDERATED_BIT_DATA
BYTE	32673	64000	-	-	-	-	BLOB	-	-	-
BYTEINT	-	-	-	-	-	-	SMALLINT	-	-	-
CHAR	1	254	-	-	-	-	CHARACTER	-	-	-
CHAR	255	32672	-	-	-	-	VARCHAR	-	-	-
CHAR	32673	64000	-	-	-	-	CLOB	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
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	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	-	-
VARBYTE	1	32762	-	-	-	-	VARCHAR	-	-	Y
VARBYTE	32763	64000	-	-	-	-	BLOB	-	-	-
VARCHAR	1	32672	-	-	-	-	VARCHAR	-	-	-
VARCHAR	32673	64000	-	-	-	-	CLOB	-	-	-
VARGRAPHIC	1	16336	-	-	-	-	VARGRAPHIC	-	-	-
VARGRAPHIC	16337	32000	-	-	-	-	DBCLOB	-	-	-

関連概念:

- 「フェデレーテッド・システム・ガイド」の『順方向データ・タイプ・マッピングと逆方向データ・タイプ・マッピング』

関連資料:

- 「フェデレーテッド・システム・ガイド」の『長形式データ・タイプから varchar データ・タイプへの変更』

- 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - NET8 ラッパー』
- 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - Sybase ラッパー』
- 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - ODBC ラッパー』
- 「フェデレーテッド・システム・ガイド」の『Unicode のデフォルトの順方向データ・タイプ・マッピング - Microsoft SQL Server ラッパー』

付録 I. デフォルトの逆方向データ・タイプ・マッピング

データ・ソースのデータ・タイプとフェデレーテッド・データベースのデータ・タイプ間のマッピングには、順方向タイプのマッピングと逆方向タイプのマッピングの 2 種類があります。順方向タイプ・マッピング では、マッピングはリモートのタイプから対応するローカル・タイプへのマッピングです。マッピングのもう一方のタイプは逆方向タイプ・マッピング であり、これはリモートの表を作成または変更するために、透過 DDL で使用されます。

ほとんどのデータ・ソースの場合、ラッパー内にデフォルトのタイプ・マッピングがあります。DB2 ファミリーのデータ・ソース用のデフォルトのタイプ・マッピングは、DRDA ラッパーにあります。Informix 用のデフォルトのタイプ・マッピングは INFORMIX ラッパーなどにあります。

DB2 フェデレーテッド・データベースにリモートの表またはビューを定義する際、その定義には、逆方向タイプ・マッピングが含まれます。このマッピングは、それぞれの列のローカル の DB2 for Linux, UNIX, and Windows のデータ・タイプからのもの、および対応するリモート・データ・タイプからのものです。たとえば、ローカル・タイプ REAL が Informix タイプ SMALLFLOAT を指す、デフォルトの逆方向タイプ・マッピングがあります。

DB2 for Linux, UNIX, and Windows フェデレーテッド・サーバーは、LONG VARCHAR、LONG VARGRAPHIC、DATALINK、およびユーザー定義タイプのマッピングをサポートしていません。

CREATE TABLE ステートメントを使用してリモート表を作成する時に、リモート表に含めたいローカル・データ・タイプを指定します。これらのデフォルトの逆方向タイプ・マッピングは、これらの列に対応するリモート・タイプを割り当てます。たとえば、CREATE TABLE ステートメントを使用して、列 C2 を持つ Informix の表を定義するとします。ステートメント内で C2 のデータ・タイプとして BIGINT を指定します。BIGINT のデフォルトの逆方向タイプ・マッピングは、どのバージョンの Informix で表を作成しているかにより異なります。Informix バージョン 8 では Informix 表の C2 のマッピングは DECIMAL となり、Informix バージョン 9 では INT8 になります。

デフォルトの逆方向タイプ・マッピングをオーバーライドすることも、CREATE TYPE MAPPING ステートメントを使用して新しい逆方向タイプ・マッピングを作成することもできます。

次の表は、DB2 for Linux, UNIX, and Windows のローカル・データ・タイプとリモート・データ・ソースのデータ・タイプ間のデフォルトの逆方向マッピングを示しています。

これらのマッピングは、特に記述のないかぎり、サポート対象のすべてのバージョンで有効です。

DB2 for z/OS and OS/390 データ・ソース

表 165. DB2 for z/OS and OS/390 のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
I DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
I REAL	-	4	-	-	-	-	REAL	-	-	-
I SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	N

DB2 for iSeries データ・ソース

表 166. DB2 for iSeries のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	N
CHARACTER	-	-	-	-	Y	-	CHARACTER	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	NUMERIC	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPHIC	-	-	-	-	-	-	VARG	-	-	N

DB2 for VM and VSE データ・ソース

表 167. DB2 for VM and VSE のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
I REAL	-	4	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPH	-	-	N

DB2 for Linux、UNIX、および Windows のデータ・ソース

表 168. DB2 for Linux, UNIX, and Windows のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	FEDERATED_BIT_DATA
BIGINT	-	8	-	-	-	-	BIGINT	-	-	-
BLOB	-	-	-	-	-	-	BLOB	-	-	-
CHARACTER	-	-	-	-	-	-	CHAR	-	-	N
CHARACTER	-	-	-	-	Y	-	CHAR	-	-	Y
CLOB	-	-	-	-	-	-	CLOB	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DBCLOB	-	-	-	-	-	-	DBCLOB	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	DOUBLE	-	-	-
FLOAT	-	8	-	-	-	-	DOUBLE	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	N
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	REAL	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	10	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	N
VARCHAR	-	-	-	-	Y	-	VARCHAR	-	-	Y
VARGRAPH	-	-	-	-	-	-	VARGRAPHIC	-	-	N
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

Informix データ・ソース

表 169. Informix のデフォルトの逆方向データ・タイプ・マッピング

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BIGINT ¹	-	-	-	-	-	-	DECIMAL	19	-	-
BIGINT ²	-	-	-	-	-	-	INT8	-	-	-
BLOB	1	2147483647	-	-	-	-	BYTE	-	-	-
CHARACTER	-	-	-	-	N	-	CHAR	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB	1	2147483647	-	-	-	-	TEXT	-	-	-
DATE	-	4	-	-	-	-	DATE	-	-	-
DECIMAL	-	-	-	-	-	-	DECIMAL	-	-	-
DOUBLE	-	8	-	-	-	-	FLOAT	-	-	-
INTEGER	-	4	-	-	-	-	INTEGER	-	-	-
REAL	-	4	-	-	-	-	SMALLFLOAT	-	-	-
SMALLINT	-	2	-	-	-	-	SMALLINT	-	-	-
TIME	-	3	-	-	-	-	DATETIME	6	10	-
TIMESTAMP	-	10	-	-	-	-	DATETIME	0	15	-
VARCHAR	1	254	-	-	N	-	VARCHAR	-	-	-
VARCHAR	255	32672	-	-	N	-	TEXT	-	-	-
VARCHAR	-	-	-	-	Y	-	BYTE	-	-	-
VARCHAR ²	255	2048	-	-	N	-	LVARCHAR	-	-	-
VARCHAR ²	2049	32672	-	-	N	-	TEXT	-	-	-

注:

1. このタイプ・マッピングは Informix サーバーのバージョン 8 以下でのみ有効。
2. このタイプ・マッピングは Informix サーバーのバージョン 9 でのみ有効。

Informix DATETIME データ・タイプでは、DB2 UNIX and Windows フェデレーテッド・サーバーは、Informix 高水準修飾子を REMOTE_LENGTH として、Informix 低水準修飾子を REMOTE_SCALE として使用します。

Informix 修飾子は、Informix クライアント SDK datatype.h ファイルに定義されている "TU_" 定数です。定数は次のとおりです。

0 = YEAR	8 = MINUTE	13 = FRACTION(3)
2 = MONTH	10 = SECOND	14 = FRACTION(4)
4 = DAY	11 = FRACTION(1)	15 = FRACTION(5)

表 169. Informix のデフォルトの逆方向データ・タイプ・マッピング (続き)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
6 = HOUR										
			12 = FRACTION(2)							

Microsoft SQL Server データ・ソース

表 170. Microsoft SQL Server のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BIGINT ¹	-	-	-	-	-	-	bigint	-	-	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	4	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	8	-	-	-	-	float	-	-	-
INTEGER	-	-	-	-	-	-	int	-	-	-
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
REAL	-	4	-	-	-	-	real	-	-	-
TIME	-	3	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	10	-	-	-	-	datetime	-	-	-
VARCHAR	1	8000	-	-	N	-	varchar	-	-	-
VARCHAR	8001	32672	-	-	N	-	text	-	-	-
VARCHAR	1	8000	-	-	Y	-	varbinary	-	-	-

表 170. Microsoft SQL Server のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります) (続き)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
VARCHAR	8001	32672	-	-	Y	-	image	-	-	-

注:

1. このタイプ・マッピングは Microsoft SQL Server バージョン 2000 でのみ有効です。

Oracle NET8 データ・ソース

表 171. Oracle NET8 のデフォルトの逆方向データ・タイプ・マッピング

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BLOB	0	2147483647	0	0	Y	\0	BLOB	0	0	Y
CHARACTER	1	254	0	0	N	\0	CHAR	0	0	N
CHARACTER	1	254	0	0	Y	\0	RAW	0	0	Y
CLOB	0	2147483647	0	0	N	\0	CLOB	0	0	N
DATE	0	4	0	0	N	\0	DATE	0	0	N
DECIMAL	0	0	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	9	0	N
REAL	0	4	0	0	N	\0	FLOAT	63	0	N
SMALLINT	0	2	0	0	N	\0	NUMBER	4	0	N
TIME	0	3	0	0	N	\0	DATE	0	0	N
TIMESTAMP	0	10	0	0	N	\0	DATE	0	0	N

表 171. Oracle NET8 のデフォルトの逆方向データ・タイプ・マッピング (続き)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
VARCHAR	1	4000	0	0	N	\0	VARCHAR2	0	0	N
VARCHAR	1	2000	0	0	Y	\0	RAW	0	0	Y

注: DB2 Universal Database for Linux, UNIX, and Windows の BIGINT データ・タイプは透過 DDL では使用できません。リモート Oracle 表を作成する時に CREATE TABLE ステートメントで BIGINT データ・タイプを指定することはできません。

Sybase データ・ソース

表 172. Sybase CTLIB のデフォルトの逆方向データ・タイプ・マッピング

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
BIGINT	-	-	-	-	-	-	decimal	19	0	-
BLOB	-	-	-	-	-	-	image	-	-	-
CHARACTER	-	-	-	-	N	-	char	-	-	-
CHARACTER	-	-	-	-	Y	-	binary	-	-	-
CLOB	-	-	-	-	-	-	text	-	-	-
DATE	-	-	-	-	-	-	datetime	-	-	-
DECIMAL	-	-	-	-	-	-	decimal	-	-	-
DOUBLE	-	-	-	-	-	-	float	-	-	-
GRAPHIC	-	-	-	-	-	-	unichar	-	-	-
VARGRAPHIC	-	-	-	-	-	-	univarchar	-	-	-
INTEGER	-	-	-	-	-	-	integer	-	-	-
REAL	-	-	-	-	-	-	real	-	-	-

表 172. Sybase CTLIB のデフォルトの逆方向データ・タイプ・マッピング (続き)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	REMOTE_BIT_DATA
SMALLINT	-	-	-	-	-	-	smallint	-	-	-
TIME	-	-	-	-	-	-	datetime	-	-	-
TIMESTAMP	-	-	-	-	-	-	datetime	-	-	-
VARCHAR ¹	1	255	-	-	N	-	varchar	-	-	-
VARCHAR ¹	256	32672	-	-	N	-	text	-	-	-
VARCHAR ²	1	16384	-	-	N	-	varchar	-	-	-
VARCHAR ²	16385	32672	-	-	N	-	text	-	-	-
VARCHAR ¹	1	255	-	-	Y	-	varbinary	-	-	-
VARCHAR ¹	256	32672	-	-	Y	-	image	-	-	-
VARCHAR ²	1	16384	-	-	Y	-	varbinary	-	-	-
VARCHAR ²	16385	32672	-	-	Y	-	image	-	-	-

注:

- このタイプ・マッピングは、Sybase サーバーのバージョン 12.0 またはそれ以前で CTLIB を使用する場合にのみ有効です。
- このタイプ・マッピングは、Sybase サーバーのバージョン 12.5 またはそれ以降で CTLIB を使用する場合にのみ有効です。

Teradata データ・ソース

表 173. Teradata のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	FEDERATED_BIT_DATA
BLOB ¹	1	64000	-	-	-	-	VARBYTE	-	-	-

表 173. Teradata のデフォルトの逆方向データ・タイプ・マッピング (表示されていない列があります) (続き)

FEDERATED_TYPENAME	FEDERATED_LOWER_LEN	FEDERATED_UPPER_LEN	FEDERATED_LOWER_SCALE	FEDERATED_UPPER_SCALE	FEDERATED_BIT_DATA	FEDERATED_DATA_OPERATORS	REMOTE_TYPENAME	REMOTE_LENGTH	REMOTE_SCALE	FEDERATED_BIT_DATA
CHARACTER	-	-	-	-	-	-	CHARACTER	-	-	-
CHARACTER	-	-	-	-	Y	-	BYTE	-	-	-
CLOB ²	1	64000	-	-	-	-	VARCHAR	-	-	-
DATE	-	-	-	-	-	-	DATE	-	-	-
DBCLOB ³	1	32000	-	-	-	-	VARGRAPHIC	-	-	-
DECIMAL	1	18	0	18	-	-	DECIMAL	-	-	-
DECIMAL	19	31	0	31	-	-	FLOAT	-	-	-
DOUBLE	-	-	-	-	-	-	FLOAT	-	-	-
GRAPHIC	-	-	-	-	-	-	GRAPHIC	-	-	-
INTEGER	-	-	-	-	-	-	INTEGER	-	-	-
REAL	-	-	-	-	-	-	FLOAT	-	-	-
SMALLINT	-	-	-	-	-	-	SMALLINT	-	-	-
TIME	-	-	-	-	-	-	TIME	-	-	-
TIMESTAMP	-	-	-	-	-	-	TIMESTAMP	-	-	-
VARCHAR	-	-	-	-	-	-	VARCHAR	-	-	-
VARCHAR	-	-	-	-	Y	-	VARBYTE	-	-	-
VARGRAPHIC	-	-	-	-	-	-	VARGRAPHIC	-	-	-

注:

1. Teradata VARBYTE データ・タイプには、指定した長さ (1 から 64000) の DB2 BLOB データ・タイプのみ含めることができます。
2. Teradata VARCHAR データ・タイプには、指定した長さ (1 から 64000) の DB2 CLOB データ・タイプのみ含めることができます。
3. Teradata VARGRAPHIC データ・タイプには、指定した長さ (1 から 32000) の DB2 DBCLOB データ・タイプのみ含めることができます。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『順方向データ・タイプ・マッピングと逆方向データ・タイプ・マッピング』

付録 J. フェデレーテッド・システムの関数マッピング・オプション

DB2 Information Integrator は、既存の組み込みデータ・ソース関数と組み込み DB2 関数間のデフォルトのマッピングを備えています。ほとんどのデータ・ソースの場合、ラッパー内にデフォルトの関数マッピングがあります。フェデレーテッド・サーバーが認識しないデータ・ソース関数を使用するには、データ・ソース関数と、フェデレーテッド・データベースでの対応する関数との間に、関数マッピングを作成する必要があります。

関数マッピング・オプションの主目的は、データ・ソースでデータ・ソース関数を実行した場合の潜在的なコストについての情報を提供することです。プッシュダウン分析により、データ・ソースの関数が照会内にある関数を実行できるかどうかを判別します。照会オプティマイザーは、関数処理をデータ・ソースにプッシュダウンした場合、最小のコストで済むかどうかを判断します。

関数マッピング定義で提供される統計情報は、オプティマイザーが、データ・ソース関数を実行するコスト見積もりを、DB2 関数を実行するコスト見積もりと比較するのに役立ちます。

表 174. 関数マッピング・オプションおよびその設定値

オプション	有効な設定値	デフォルトの設定値
DISABLE	デフォルト関数マッピングを使用不可にする。有効な値は 'Y' および 'N'。	'N'
INITIAL_INSTS	データ・ソース関数が呼び出された最初の時および最後の時に処理される命令数の推定値	'0'
INITIAL_IOS	データ・ソース関数が呼び出された最初の時および最後の時に実行される入出力数の推定値	'0'
IOS_PER_ARGBYTE	データ・ソース関数に渡された引き数セットの、それぞれのバイトごとに費やされる入出力数の見積もり	'0'
IOS_PER_INVOC	データ・ソース関数の呼び出しごとの入出力数の推定値	'0'
INSTS_PER_ARGBYTE	データ・ソース関数に渡された引き数セットの、それぞれのバイトごとに処理される命令数の見積もり	'0'
INSTS_PER_INVOC	データ・ソース関数の呼び出しごとに処理される命令数の推定値	'450'
PERCENT_ARGBYTES	データ・ソース関数が実際に読み取る入力引き数バイトの平均パーセントの推定値	'100'
REMOTE_NAME	データ・ソース関数の名前。	ローカル名

DB2 Information Integrator technical 技術文書

以下のトピックについて説明します。

- DB2 Information Integrator PDF 資料
- DB2 Information Integrator リリース情報およびインストール要件
- DB2 Information Integrator 資料のフィックスパック

DB2 Information Integrator の資料

DB2 PDF ドキュメンテーション CD には、DB2 Information Integrator ライブラリーと DB2 Universal Database ライブラリーの資料の PDF ファイルが収録されています。DB2 PDF ドキュメンテーション CD の構造は、以下のようになっています。

- Windows オペレーティング・システムの場合: `x:\%doc%\%L`
- UNIX オペレーティング・システムの場合: `/cdrom/doc/%L/`

各部分の意味は以下のとおりです。

- `x` は Windows の CD ドライブ名を表します。
- `cdrom` は CD のマウント・ポイントを表します (UNIX)。
- `%L` は使用する資料のロケールです (en_US など)。

次の表には、PDF 文書へのアクセスに使用できる言語 ID がリストされています。

表 175. 言語ロケールおよび ID

言語	ロケール	ID	言語	ロケール	ID
アラビア語	ar_AA	w	日本語	ja_JP	j
ブラジル・ポルトガル語	pt_BR	b	韓国語	ko_KR	k
ブルガリア語	bg_BG	u	ノルウェー語	no_NO	n
クロアチア語	hr_HR	9	ポーランド語	pl_PL	p
チェコ語	cs_CZ	x	ポルトガル語	pt_PT	v
デンマーク語	da_DK	d	ルーマニア語	ro_RO	8
オランダ語	nl_NL	q	ロシア語	ru_RU	r
英語	en_US	e	中国語 (簡体字)	zh_CN	c
フィンランド語	fi_FI	y	スロバキア語	sk_SK	7
フランス語	fr_FR	f	スロベニア語	sl_SI	l
ドイツ語	de_DE	g	スペイン語	es_ES	z
ギリシャ語	el_GR	a	スウェーデン語	sv_SE	s
ハンガリー語	hu_HU	h	中国語 (繁体字)	zh_TW	t
イタリア語	it_IT	i	トルコ語	tr_TR	m

各 PDF ファイル名の 6 番目の文字は資料の言語を示しています (以下の表をご覧ください)。たとえば、ファイル名 `iiyige80` は「*IBM DB2 Information Integrator* インストール・ガイド」の英語版を示しており、ファイル名 `iiyigg80` は同じ資料のドイツ語版を示しています。

次の表には、DB2 Information Integrator 用の利用可能な PDF 資料がリストされています。

表 176. DB2 Information Integrator 資料

資料名	資料番号	PDF ファイル名
<i>IBM DB2 Information Integrator</i> アプリケーション開発者向けガイド	SC88-9609	iiyfsx81
<i>IBM DB2 Information Integrator</i> ラッパー開発における C++ API リファレンス	SC88-9921	iiywx80
<i>IBM DB2 Information Integrator</i> データ・ソース構成ガイド	オンラインでのみ入手可能	
<i>IBM DB2 Information Integrator</i> フェデレーテッド・システム・ガイド	SC88-9164	iiyfx81
<i>IBM DB2 Information Integrator</i> インストール・ガイド	GC88-9562	iiyix81
<i>IBM DB2 Information Integrator</i> レプリケーションとイベント・パブリッシング入門	GC88-9895	db2gpx80
<i>IBM DB2 Information Integrator</i> ラッパー開発における Java API リファレンス	SC88-9922	iiyjr80
<i>IBM DB2 Information Integrator</i> マイグレーション・ガイド	SC88-9610	iiymgx81
<i>IBM DB2 Information Integrator</i> レプリケーションとイベント・パブリッシング ガイドおよびリファレンス	SC88-9893	db2qrx80
<i>IBM DB2 Information Integrator</i> SQL レプリケーション・ガイドおよびリファレンス	SC88-9163	db2e0x82
<i>IBM DB2 Information Integrator</i> ラッパー開発者向けガイド	SC88-9923	iiywdx80

リリース情報およびインストール要件

リリース情報およびインストール要件では、製品のリリースおよびフィックスパックのレベルに固有の情報を提供します。また、各リリースおよびフィックスパックに組み込まれているドキュメンテーション更新のサマリーも提供します。

リリース情報およびインストール要件は、テキスト形式および HTML 形式で製品 CD から入手できます。

- Windows オペレーティング・システムの場合: `x:%doc%L`
- UNIX オペレーティング・システムの場合: `/cdrom/doc/%L/`

各部分の意味は以下のとおりです。

- `x` は Windows の CD ドライブ名を表します。
- `cdrom` は CD のマウント・ポイントを表します (UNIX)。
- `%L` は使用する資料のロケールです (`en_US` など)。

次の表は、リリース情報およびインストール要件のロケーションを示しています。

表 177. リリース情報およびインストール要件

資料名	ファイル名	位置
<i>IBM DB2 Information Integrator</i> リリース・ノート	ReleaseNotes	<ul style="list-style-type: none">DB2 インフォメーション・センターにあるリリース情報Web 上で http://www.ibm.com/software/data/integration/db2ii/support.html にアクセスします。
<i>IBM DB2 Information Integrator</i> インストール要件	Prereqs	<ul style="list-style-type: none">製品 CDDB2 Information Integration Installation Launchpad

DB2 Information Integrator ドキュメンテーション・フィックスパック

IBM は、DB2 インフォメーション・センターの資料のフィックスパックやその他の資料更新を定期的に発行しています。DB2 インフォメーション・センター (<http://publib.boulder.ibm.com/infocenter/db2help/>) にアクセスすれば、常に最新の情報が掲載されます。DB2 インフォメーション・センターをローカル・インストールしている場合、更新記事を表示するには、まず手動で更新をインストールしてください。新しい情報が発表されたときに資料を更新することにより、*DB2* インフォメーション・センター CD からインストールした情報を更新することができます。

インフォメーション・センターの方が、PDF 資料やハードコピー資料よりも頻繁に更新されます。DB2 の最新の技術情報を入手するには、資料更新が発行されたときにそれをインストールするか、または www.ibm.com サイトの DB2 インフォメーション・センターにアクセスしてください。

アクセス支援

アクセス支援機能は、身体に障害のある（身体動作が制限されている、視力が弱いなど）ユーザーがソフトウェア製品を十分活用できるように支援します。DB2[®]バージョン 8 製品に備わっている主なアクセス支援機能は、以下のとおりです。

- すべての DB2 機能は、マウスの代わりにキーボードを使ってナビゲーションできます。詳細については、『キーボードによる入力およびナビゲーション』を参照してください。
- DB2 ユーザー・インターフェースのフォント・サイズおよび色をカスタマイズすることができます。詳細については、704 ページの『アクセスしやすい表示』を参照してください。
- DB2 製品は、Java[™] Accessibility API を使用するアクセス支援アプリケーションをサポートします。詳細については、704 ページの『支援テクノロジーとの互換性』を参照してください。
- DB2 資料は、アクセスしやすい形式で提供されています。詳細については、704 ページの『アクセスしやすい資料』を参照してください。

キーボードによる入力およびナビゲーション

キーボード入力

キーボードだけを使用して DB2 ツールを操作できます。マウスを使って実行できる操作は、キーまたはキーの組み合わせによっても実行できます。標準のオペレーティング・システム・キー・ストロークを使用して、標準のオペレーティング・システム操作を実行できます。

キーまたはキーの組み合わせによって操作を実行する方法について、詳しくは キーボード・ショートカットおよびアクセラレーター: Common GUI help を参照してください。

キーボード・ナビゲーション

キーまたはキーの組み合わせを使用して、DB2 ツールのユーザー・インターフェースをナビゲートできます。

キーまたはキーの組み合わせによって DB2 ツールをナビゲートする方法の詳細については、キーボード・ショートカットおよびアクセラレーター: Common GUI help を参照してください。

キーボード・フォーカス

UNIX[®] オペレーティング・システムでは、アクティブ・ウィンドウの中で、キー・ストロークによって操作できる領域が強調表示されます。

アクセスしやすい表示

DB2 ツールには、視力の弱いユーザー、その他の視力障害をもつユーザーのためにアクセシビリティを向上させる機能が備わっています。これらのアクセシビリティ拡張機能には、フォント・プロパティのカスタマイズを可能にする機能も含まれています。

フォントの設定

「ツール設定」ノートブックを使用して、メニューおよびダイアログ・ウィンドウに使用されるテキストの色、サイズ、およびフォントを選択できます。

フォント設定に関する詳細情報は、メニューおよびテキストのフォントを変更する: [Common GUI help](#) を参照してください。

色に依存しない

本製品のすべての機能を使用するために、ユーザーは必ずしも色を識別する必要はありません。

支援テクノロジーとの互換性

DB2 ツールのインターフェースは、Java Accessibility API をサポートします。これによって、スクリーン・リーダーその他の支援テクノロジーを DB2 製品で利用できるようになります。

アクセスしやすい資料

DB2 形式は、ほとんどの Web ブラウザーで表示可能な XHTML 1.0 形式で提供されています。XHTML により、ご使用のブラウザに設定されている表示設定に従って資料を表示できます。さらに、スクリーン・リーダーや他の支援テクノロジーを使用することもできます。

シンタックス・ダイアグラムはドット 10 進形式で提供されます。この形式は、スクリーン・リーダーを使用してオンライン・ドキュメンテーションにアクセスする場合にのみ使用できます。

関連概念:

- ・ [インフラストラクチャー・トピック \(DB2 共通ファイル\)の『ドット 10 進シンタックス・ダイアグラム』](#)

関連タスク:

- ・ [『キーボード・ショートカットおよびアクセラレーター: Common GUI help』](#)
- ・ [『メニューおよびテキストのフォントを変更する: Common GUI help』](#)

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケ

ーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

以下は、IBM Corporation の商標です。

IBM
DB2
DB2 Universal Database
Domino
Domino.Doc
Informix
Lotus
Lotus Discovery Server
Lotus Notes
QuickPlace
Sametime
SecureWay
WebSphere

以下は、それぞれ各社の商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標および登録商標です。

Microsoft、Windows、Windows NT、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

MMX および Pentium は、Intel Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセシビリティ
機能 703
アクセス・プラン
説明 10
アプリケーション・データ・エンティティ
ー
ビジネス・オブジェクト定義 137
エラーの検査
Web サービス・ラッパー 460
大文字小文字の区別
大文字小文字の区別の保持 26
フェデレーテッド・データ・ソースの
チェックリスト 35
オブジェクト
ネーミング 25
オプション
ニックネーム 649
オプティマイザー
説明 10

[カ行]

外部キー
ニックネームとの使用 156
カスタム関数
BioRS 79, 80, 87
Documentum 204
Entrez 219, 220, 226
Extended Search 269
カタログ
「グローバル・カタログ」を参照 617
環境変数
フェデレーテッドでの概要 65
フェデレーテッド複数パーティシヨ
ン・インスタンス内の 67
Documentum 193
Informix 309
Microsoft SQL Server 328
Oracle 366
Sybase 382
Teradata 412
関数テンプレート
Extended Search 269

関数マッピング
オプション
有効な設定値 697
説明 21
プラン 35
キーボード・ショートカット
サポート 703
疑似列
Documentum 198
逆 UDF
LSRevComp 524
LSRevNuc 525
LSRevPep 526
逆変換 UDF
概要 500
LSPep2AmbNuc 500
LSPep2ProbNuc 503
逆方向タイプ・マッピング
デフォルト・マッピング 685
許可
表構造ファイルのファイル・アクセス
404
Excel ファイルのファイル・アクセス
250
組み込み関数 21
クライアント・ライブラリー
Documentum 192
繰り返し列属性
Documentum 203
グローバル・カタログ
説明 9
フェデレーテッド情報を含むビュー
617
コード・ページ 55
フェデレーテッド・システム 52
更新
統計
プラン 27
高スコア・セグメント・ペア 107
構成
フェデレーテッド・データ・ソース
概要 74
構文
KEGG ユーザー定義関数 537
固定列
BLAST ニックネーム 123
HMMER ニックネーム 297
コドン度数表 531, 532
コネクター・コンフィギュレーター
ビジネス・オブジェクト・アプリケー
ション構成 139

コマンド行プロセッサ (CLP)
フェデレーテッド関数 22
コマンド・センター
データ・ソースの構成 61
フェデレーテッドでの使用 22
コントロール・センター
データ・ソースの構成 61, 63
フェデレーテッド・システム用のイン
ターフェース 22

[サ行]

サーバー定義
説明 15
登録の概要 69
ビジネス・アプリケーション 145
表構造ファイル 402
BioRS 82
BLAST 121
DB2 ファミリー・データ・ソース
180
Documentum 195
Entrez 222
Excel ファイル 247
Extended Search 266
HMMER 294
Informix 315
Microsoft SQL Server 332
ODBC 346
OLE DB 361
Oracle 373
Sybase 387
Teradata 416
Web サービス 435
WebSphere Business Integration ラッパ
ー 145
XML 469
サーバー・オプション
一時 15
説明 15
有効な設定値 629
サーバー・タイプ
有効なフェデレーテッド・タイプ 623
最適化
表構造ファイル 405
BLAST 132
索引の指定
説明 21
サンプル
照会
Extended Search 270

シナリオ
Excel ファイルへのアクセス 249
主キー
ニックネームとの使用 156
述部
Entrez 227
順方向タイプ・マッピング
デフォルト・マッピング 669
照会
データ・ソースへのアクセス
ビジネス・アプリケーション 173
BioRS 90, 92
BLAST 129, 130
Documentum 209, 211
Entrez 228, 229
Excel 248
Extended Search 272
HMMER 304
Web サービス 456
XML 481
フラグメント 10
例
ビジネス・アプリケーション 169,
452
Web サービス・ラッパー 169,
452
XML ファイルに使用されるコスト・
モデル 473
照会の最適化
説明 10
BioRS 照会のガイドライン 97
XML 480
照合シーケンス
フェデレーテッド・システム 51, 52,
54
身体障害 703
垂直表
Extended Search 261
正規表現のサポート
ライフ・サイエンス・ユーザー定義関
数 515
制約事項
ニックネームの変更 576
db2dj.ini ファイル 66
セキュリティー
Web サービス・ラッパー 459

[タ行]

チェックリスト
フェデレーテッド・システム構成の計
画 35
チューニング
データ・ソース構成
Informix 322
Microsoft SQL Server 339

チューニング (続き)
データ・ソース構成 (続き)
ODBC 357
Oracle 379
Sybase 395
Teradata 424
データ・エレメント
Entrez のための変換 228
データ・ソース 8, 10
オプションの構成ステップ 74
構成 61
説明 5
デフォルトのラッパー名 14
ビジネス・アプリケーション 142
有効なサーバー・タイプ 623
データ・ソースの構成
ニックネーム・オプション 649
DB2 コントロール・センターの使用
63
データ・ソース・オブジェクト
説明 17
有効なオブジェクト・タイプ 18
データ・タイプ
サポートされない 20
非リレーショナル・データ・ソースの
569
データ・タイプ・マッピング
順方向 669
説明 20
特定データ・ソース・オブジェクトの
583, 584
反転 685
フェデレーテッド・データ・ソースの
チェックリスト 35
プラン 33
データ・バンク
説明 77
デーモン
構成ファイル
BLAST の例 116
HMMER の例 290
BLAST のための開始 117
BLAST のための構成 113
HMMER のための開始 290
HMMER のための構成 286
定義行の構文解析
説明 122
適合、説明 11
テンプレート
ビジネス・オブジェクト 137
ドイツ Biomax 社 77
統計
データ・ソースの更新 27
フェデレーテッド・データ・ソースの
チェックリスト 35
BioRS の場合 98, 99

トラブルシューティング
DB2 ファミリー・サーバー接続のテス
ト 184
Informix サーバー接続のテスト 319
Microsoft SQL Server 構成 339
Microsoft SQL Server 接続のテスト
336
ODBC 構成 357
ODBC サーバー接続のテスト 350
Oracle 構成 379
Oracle サーバー接続のテスト 377
Sybase 構成 395
Sybase サーバー接続のテスト 392
Teradata 構成 424
Teradata サーバー接続のテスト 410,
420
Web サービス・ラッパー 460

[ナ行]

ニックネーム
作成
拡張検索 260
ニックネームの 73
ビジネス・アプリケーション・デー
タ・ソース 146
表構造ファイル 402
BioRS 84
Entrez 224
HMMER 296
Informix 320
Microsoft SQL Server 338
ODBC 351
Oracle 378
Sybase 393
Teradata 422
Web サービス 436, 446
WebSphere Business Integration デ
ータ・ソース 146
XML 474
指定
非リレーショナル・データ・ソース
の列 74
説明 17, 154, 443
データベース技法 169, 452
登録
概要 71
ビジネス・アプリケーション 135
変更
概要 575
制約事項 576
ニックネーム・オプション 580
列オプション 581
ローカル列名 578
ローカル・データ・タイプ 583

ニックネーム (続き)
 変更 (続き)
 ローカル・データ・タイプ、例 584
 有効なデータ・ソース・オブジェクト 18
 DB2 ファミリー・データ・ソース 185
 Web サービス記述言語 (WSDL) 427
 ニックネームおよび列オプション
 Web サービス・ラッパー 149, 427, 438
 WebSphere Business Integration ラッパー 149, 438
 ニックネーム階層、例 156
 ニックネーム階層、例 156
 ニックネーム統計
 BioRS 99, 100, 101
 ニックネーム列オプション
 説明 19
 例 581
 ニックネーム・オプション
 指定 580
 ノード
 カタログ、DB2 ファミリー・データ・ソースの場合 176

[ハ行]

パススルー
 制約事項 12
 説明 12
 パスワード
 値の大文字小文字が区別される 26
 パターン・マッチング UDF
 LSPatternMatch 512
 LSPrositePattern 514
 パラメーター
 FEDERATED 49
 汎用照会言語
 Extended Search 274
 ビジネス・アプリケーション
 アダプターの構成 139
 サーバー定義 145
 サンプルの照会 173
 データ・タイプ、サポートされる 569
 フェデレーテッド・ビュー 155
 ラッパー 143
 ラッパーによるアクセス 137
 ラッパー・ライブラリー・ファイル 144
 ビジネス・アプリケーション・データ・ソース
 ニックネームの登録 146
 フェデレーテッド・システムへの追加 142

ビジネス・アプリケーション・データ・ソース (続き)
 フェデレーテッド・ビュー 445
 へのアクセスの構成 142
 ビジネス・オブジェクト 137
 フェデレーテッド・システムへの追加 142
 WebSphere Business Integration ラッパー 135
 ビジネス・オブジェクトの廃棄
 ニックネーム 135
 ビジネス・オブジェクト・アプリケーション
 WebSphere Business Integration ラッパー 135
 ビジネス・オブジェクト・アプリケーション・ラッパー
 例 169, 452
 ビュー
 ビジネス・アプリケーション 155, 445
 フェデレーテッド、非ルート・ニックネームの 479
 表構造ファイル
 概要 397
 サーバー定義の登録 402
 最適化 405
 サポートされるバージョン 6
 サンプルの照会 398
 タイプ 397
 データ・タイプ、サポートされる 569
 ニックネーム、有効なオブジェクト 18
 ニックネームの登録 402
 ファイル・アクセス制御モデル 404
 フェデレーテッド・システムへの追加 400
 へのアクセスの構成 400
 メッセージ 405
 ラッパーの登録 400
 ラッパー・ライブラリー・ファイル 401
 例 397
 CREATE NICKNAME 構文 605
 CREATE NICKNAME ステートメントの例 403
 DB2 Information Integrator によるアクセス 398
 非リレーショナル・データ・ソース
 サポートされるデータ・タイプ 569
 データ・タイプ・マッピングの指定 20
 ニックネーム列 74
 ファイル・タイプ
 表構造ファイル 397
 フェデレーテッド・サーバー 5

フェデレーテッド・サーバー (続き)
 セットアップの検査 43
 説明 4
 複数のサーバーの構成 64
 ラッパー 13
 ラッパー・モジュール 13
 フェデレーテッド・システム
 概要 3
 フェデレーテッド・データベース
 作成 57
 システム・カタログ 9
 照合シーケンス 52, 54
 説明 8
 フェデレーテッド・ビュー
 ビジネス・アプリケーション 445
 非ルート・ニックネームの場合 479
 複数パーティション構成 67
 プッシュダウン分析
 説明 10
 フラット・ファイル
 「表構造ファイル」を参照 6
 プラン
 関数マッピング 35
 サーバー定義の登録 69
 正しいラッパーの選択 28
 データ・ソース環境変数 65
 データ・タイプ・マッピング 33
 ニックネームの登録 71
 非リレーショナル・ニックネーム列 74
 フェデレーテッド・サーバー
 Microsoft SQL Server の場合 326
 ODBC のため 343
 フェデレーテッド・サーバー・セットアップの検査 43
 フェデレーテッド・システム構成のチェックリスト 35
 ユーザー・マッピング 33
 ユーザー・マッピングの登録 70
 ラッパーの登録 68
 Excel データへのアクセス 29
 分散データベース管理システム 3
 変換表
 ライフ・サイエンス・ユーザー定義関数
 形式 532
 例 533
 変更
 ニックネーム
 概要 575
 ニックネーム・オプション 580
 ローカル・データ・タイプ 583

[マ行]

マッピング

XML 文書 154, 443

マップ・フィールド

Extended Search 261

命名規則

フェデレーテッド・データベース・オブジェクト 25

フェデレーテッド・データ・ソースのチェックリスト 35

メッセージ

表構造ファイル・ラッパー 405

BioRS ラッパー 102

BLAST ラッパー 132

djxlink 46

Documentum ラッパー 211

Entrez ラッパー 239

Excel ラッパー 251

Extended Search ラッパー 276

HMMER ラッパー 305

LSPep2AmbNuc ユーザー定義関数 503

LSPep2ProbNuc ユーザー定義関数 505

Web サービス・ラッパー 460

XML ラッパー 482

[ヤ行]

ユーザー ID

値の大文字小文字が区別される 26

ユーザー定義関数

KEGG 536

ユーザー定義関数 (UDF) 21

除去、ライフ・サイエンス 499

除去、KEGG 565

登録、ライフ・サイエンス 497

登録、KEGG 538

ライフ・サイエンス 495, 496

KEGG 535

ユーザー定義タイプ (UDT)

サポートされないデータ・タイプ 20

ユーザー・マッピング

オプション 16

説明 16

登録の概要 70

フェデレーテッド・データ・ソースのチェックリスト 35

プラン 33

有効な設定値 647

Web サービス・ラッパー 459

[ラ行]

ライブラリー

複数のクライアント・ライブラリーへの手動でのリンク 47

ユーザー定義関数 495

ラッパー 44, 45

Informix ラッパー 314

Microsoft SQL Server ラッパー 331

Oracle ラッパー 372

Sybase ラッパー 387

Teradata ラッパー 416

ライフ・サイエンス・ユーザー定義関数

概要 495

概要、逆変換関数 500

概要、UDF の定義行構文解析 506

関数のリスト 496

除去 499

正規表現のサポート 515

登録 497

フォーマット、変換表 532

例、変換表 533

GeneWise 516

ラッパー

説明 13

デフォルト名 14

登録の概要 68

ビジネス・アプリケーション 135, 143

表構造ファイルの場合の登録 400

フェデレーテッド・データ・ソースのチェックリスト 35

プラン 28

ライブラリー 44

ライブラリー・ファイル 45

BioRS の場合の登録 81

BLAST の場合の登録 119

DB2 ファミリー・データ・ソースの場合の登録 178

Documentum の場合の登録 194

Entrez の場合の登録 220

Excel ファイルの場合の登録 245

Extended Search の場合の登録 265

HMMER の場合の登録 293

Informix の場合の登録 313

Microsoft SQL Server のための登録 330

ODBC の場合の登録 344

OLE DB の場合の登録 360

Oracle の場合の登録 371

Sybase の場合の登録 386

Teradata の場合の登録 415

Web サービス・データ・ソース 433, 434

WebSphere Business Integration 143

XML の場合の登録 467

ラッパーの作成

表構造ファイル 400

BioRS 81

BLAST 119

DB2 ファミリー・データ・ソース 178

Documentum 194

Entrez 220

Excel ファイル 245

Extended Search 265

HMMER 293

Informix 313

ODBC 344

OLE DB 360

Oracle 371

Sybase 386

Teradata 415

XML 467

ラッパー・オプション

有効な設定値 621

ラッパー・ライブラリー 435

表構造ファイル名 401

への手動でのリンク 47

BioRS ファイル名 82

BLAST ファイル名 120

DB2 ファミリー・ファイル名 179

Documentum ファイル名 195

Entrez ファイル名 221

Excel ファイル名 246

Extended Search ファイル名 266

HMMER ファイル名 294

Informix ファイル名 314

Microsoft SQL Server ファイル名 331

ODBC ファイル名 345

OLE DB ファイル名 361

Oracle ファイル名 372

Sybase ファイル名 387

Teradata ファイル名 416

WebSphere Business Integration ラッパー・ファイル名 144

XML ファイル名 468

リモート・カタログ情報 9

例

コドン度数表 532

ビジネス・アプリケーション・ラッパー 169, 452

db2blast_daemon コマンド 118

db2hmmer_daemon コマンド 292

Excel ファイル 29, 243

Web サービス・ラッパー 169, 446, 452

WebSphere Business Integration ラッパー 156

XML 文書マッピング 463

例、照会
 ビジネス・アプリケーション・データ・ソース 173
 表構造ファイル 398
 BioRS 90, 92
 BLAST 130
 Documentum 209
 Entrez 229
 Excel 248
 Extended Search 272
 HMMER 302
 Web サービス 456
 XML 481

例、デーモン
 BLAST 構成ファイル 116
 HMMER 構成ファイル 290

例、CREATE USER MAPPING
 Oracle 375
 Teradata 420

例、CREATE USER MAPPING ステートメント
 DB2 ファミリー・データ・ソース 183
 Informix 318
 Microsoft SQL Server 335
 ODBC 349
 OLE DB 364
 Sybase 391

例、CREATE WRAPPER ステートメント
 ODBC 345

例、UDF
 ライフ・サイエンスの変換表 533
 LSBarCode ユーザー定義関数 519
 LSDeflineParse ユーザー定義関数 511
 LSMultiMatch ユーザー定義関数 521
 LSMultiMatch3 ユーザー定義関数 522
 LSNuc2Pep ユーザー定義関数 528
 LSPatternMatch ユーザー定義関数 512
 LSPep2AmbNuc ユーザー定義関数 501
 LSPep2ProbNuc ユーザー定義関数 504
 LSPrositePattern ユーザー定義関数 514
 LSRevComp ユーザー定義関数 524
 LSRevNuc ユーザー定義関数 526
 LSRevPep ユーザー定義関数 527
 LSTransAllFrames 530

列オプション
 説明 19
 ニックネームの指定 581
 有効な設定値 661

列名
 変更 578

ローカル・カタログ
 「グローバル・カタログ」を参照 9

A

ACCOUNTING_STRING ユーザー・オプション
 有効な設定値 647

Action Output ユーティリティ
 フェデレーテッド・サーバーの構成 64

AllNbrsByGeneS ユーザー定義関数
 構文 550

AllNbrsByGeneT ユーザー定義関数
 構文 551

AllText エレメント
 BioRS 91

ALTER NICKNAME ステートメント
 制約事項 576
 例
 列オプションの変更 581
 ローカル列名の変更 578

ALTER NICKNAME ステートメント
 例
 ローカル・データ・タイプ 584

B

BestHmlgsByGenesS ユーザー定義関数
 構文 560

BestHmlgsByGenesT ユーザー定義関数
 構文 560

BestNbrsByGeneS ユーザー定義関数
 構文 554

BestNbrsByGeneT ユーザー定義関数
 構文 555

BioRS
 カスタム関数 79, 87
 カスタム関数の登録 80
 結合述部の使用 90
 サーバー定義の登録 82
 サンプルの照会 90, 92
 説明 77
 データ・タイプ、サポートされる 569
 統計
 データ・バンク・カーディナリティー 99
 ニックネーム・カーディナリティー 99
 保守 98
 列カーディナリティー 100, 101
 ニックネームの登録 84
 フェデレーテッド・システムへの追加 78
 へのアクセスの構成 78

BioRS (続き)

メッセージ 102
 ラッパーの登録 81
 ラッパー・パフォーマンスの最適化 97
 ラッパー・ライブラリー・ファイル 82
 AllText エレメント 91
 CREATE NICKNAME 構文 589
 CREATE NICKNAME ステートメントの例 86
 CREATE SERVER ステートメント 587
 CREATE USER MAPPING ステートメント 83, 588

BLAST
 高スコア・セグメント・ペア (HSP) 107
 構成の最適化 132
 固定入出力列 123
 サーバー定義の登録 121
 サポートされるバージョン 6
 サンプルの照会 130
 照会 129
 データ・タイプ、サポートされる 569
 デーモン
 開始 117
 構成 113
 デーモン構成ファイル
 例 116
 定義行の構文解析 122
 ニックネーム、有効なオブジェクト 18
 ニックネームの登録 121
 フェデレーテッド・システムへの追加 111
 正しい Blast の実行可能なものが
 がインストールされていることを
 検査 112
 正しいマトリックス・ファイルがイ
 ンストールされていることを検査
 112
 へのアクセスの構成 111
 メッセージ 132
 ラッパーの登録 119
 ラッパー・ライブラリー・ファイル 120
 CREATE NICKNAME 構文 591
 CREATE NICKNAME ステートメント 121
 CREATE NICKNAME ステートメント
 の例 127
 CREATE SERVER ステートメント 590
 TurboBlast の構成 128

BLASTALL_PATH
BLAST デーモン 113
BLAST_OUT_DIR_PATH
BLAST デーモン 113
BstBstHmlgByGenesS ユーザー定義関数
構文 561
BstBstHmlgByGenesT ユーザー定義関数
構文 562
BstBstNbrsByGeneS ユーザー定義関数
構文 552
BstBstNbrsByGeneT ユーザー定義関数
構文 553

C

CATALOG DATABASE コマンド
DB2 ファミリー・データ・ソースへの
アクセス 177
CATALOG TCP/IP NODE コマンド
DB2 ファミリー・データ・ソースへの
アクセス 176
CLP (コマンド行プロセッサ)
フェデレーテッド関数 22
COLLATING_SEQUENCE サーバー・オブ
ション
有効な設定値 629
COMM_RATE サーバー・オプション
有効な設定値 629
CompoundsByPathwyS ユーザー定義関数
構文 541
CompoundsByPathwyT ユーザー定義関数
構文 542
CONNECTSTRING サーバー・オプション
有効な設定値 629
CPU_RATIO サーバー・オプション
有効な設定値 629
CREATE FUNCTION ステートメント
Extended Search 269, 604
CREATE INDEX ステートメント 21
CREATE NICKNAME 構文
表構造ファイル 605
BioRS 589
BLAST 591
Documentum 594
Excel 597
Extended Search 601
XML 610
CREATE NICKNAME ステートメント
73
ビジネス・アプリケーション・デー
タ・ソース 146
表構造ファイル 402, 403
BioRS 84, 86
BLAST 121, 127
DB2 ファミリー・データ・ソース
185, 186

CREATE NICKNAME ステートメント
(続き)
Documentum 197, 202
Entrez 224, 605
Excel ファイル 247
Extended Search 268
HMMER 296, 301
Informix 320, 321
Microsoft SQL Server 338, 339
ODBC 351, 353
Oracle 378
Sybase 393, 394
Teradata 422, 423
Web サービス 436
WebSphere Business Integration ラッパ
ー 146
XML 474
CREATE SERVER ステートメント 4
表構造ファイル 402
BioRS 82, 587
BLAST 121, 590
DB2 ファミリー・データ・ソース
180, 181
Documentum 195, 592
Entrez 222, 604
Excel 597
Excel ファイル 247
Extended Search 266, 599
HMMER 295
Informix 315, 316
Microsoft SQL Server 332, 333
ODBC 346, 347
OLE DB 361, 362
Oracle 373, 374
Sybase 387, 389
Teradata 416, 418
Web サービス 435
XML 469
CREATE USER MAPPING ステートメン
ト
BioRS 83, 588
DB2 ファミリー・データ・ソース
182, 183
Documentum 196, 593
Extended Search 267, 601
Informix 318
Microsoft SQL Server 335
ODBC 348, 349
OLE DB 363, 364
Oracle 375
Sybase 391
Teradata 419, 420
CREATE WRAPPER ステートメント
Extended Search 599
ODBC 345

D

DAEMON_LOGFILE_DIR
BLAST デーモン 113
HMMER デーモン 286
DAEMON_PORT
BLAST デーモン 113
HMMER デーモン 286
DATALINK データ・タイプ
サポートされない 20
DATEFORMAT サーバー・オプション
有効な設定値 629
DB2 for iSeries
サポートされるバージョン 6
デフォルトの逆方向タイプ・マッピ
ング 685
デフォルトの順方向タイプ・マッピ
ング 669
デフォルトのラッパー名 14
ニックネーム、有効なオブジェクト
18
有効なサーバー・タイプ 623
DB2 for Linux, UNIX, Windows
サポートされるバージョン 6
デフォルトの逆方向タイプ・マッピ
ング 685
デフォルトの順方向タイプ・マッピ
ング 669
デフォルトのラッパー名 14
ニックネーム、有効なオブジェクト
18
有効なサーバー・タイプ 623
DB2 for VM and VSE
サポートされるバージョン 6
デフォルトの逆方向タイプ・マッピ
ング 685
デフォルトの順方向タイプ・マッピ
ング 669
デフォルトのラッパー名 14
ニックネーム、有効なオブジェクト
18
有効なサーバー・タイプ 623
DB2 for z/OS and OS/390
サポートされるバージョン 6
デフォルトの逆方向タイプ・マッピ
ング 685
デフォルトの順方向タイプ・マッピ
ング 669
デフォルトのラッパー名 14
ニックネーム、有効なオブジェクト
18
有効なサーバー・タイプ 623
DB2 ファミリー・データ・ソース
サーバー接続のテスト 184
サーバー定義の登録 180
ニックネームの登録 185

DB2 ファミリー・データ・ソース (続き)
ノード項目のカタログ 176
フェデレーテッド・システムへの追加
175
へのアクセスの構成 175
ユーザー・マッピングの登録 182
ラッパーの登録 178
ラッパー・ライブラリー・ファイル
179
リモート・データベースのカタログ
177
CREATE NICKNAME ステートメント
の例 186
CREATE SERVER ステートメント
181
CREATE USER MAPPING ステートメ
ント 183
db2blast_daemon コマンド 118
発行するステップ 117
db2dj.ini
制約事項 66
複数パーティション・インスタンス構
成 67
db2hmmmer_daemon コマンド
発行するステップ 290
例 292
db2nodes.cfg ファイル
フェデレーテッド・データベース 57
DB2_DJ_COMM 環境変数
Informix、チューニング 322
Microsoft SQL Server、チューニング
339
ODBC、チューニング 357
Oracle、チューニング 379
Sybase、チューニング 395
Teradata のチューニング 424
DB2_MAXIMAL_PUSHDOWN サーバー・
オプション
有効な設定値 629
DBNAME サーバー・オプション
有効な設定値 629
DefinitionsByGeneS ユーザー定義関数
構文 563
DISABLE function 関数マッピング・オブ
ション
有効な設定値 697
djlxlk
エラー・メッセージ・ファイル 46
クライアント・ライブラリーへの手動
でのリンク 47
クライアント・ライブラリーへのリン
クの確認 43
dmcl.ini ファイル
Documentum 193
Documentum
カスタム関数 204

Documentum (続き)
環境変数、設定 193
疑似列 198
クライアント・ライブラリー 192
繰り返し属性の二重の定義 203
サーバー定義の登録 195
サポートされるバージョン 6
サンプルの照会 209
資料へのユーザー・アクセス 211
説明 189
データ・タイプ、サポートされる 569
ニックネーム、有効なオブジェクト
18
ニックネームの登録 197
ニックネーム列 203
フェデレーテッド・システムへの追加
191
へのアクセスの構成 191
メッセージ 211
ユーザー・マッピングの登録 196
ラッパーの登録 194
ラッパー・ライブラリー・ファイル
195
例 189
CREATE NICKNAME 構文 594
CREATE NICKNAME ステートメント
197
CREATE NICKNAME ステートメント
の例 202
CREATE SERVER ステートメント
592
CREATE USER MAPPING ステートメ
ント 196、593

E

Entrez
カスタム関数 219、226
カスタム関数の登録 220
関係述部 227
サーバー定義の登録 222
サポートされるバージョン 6
サンプルの照会 229
説明 217
ニックネーム、有効なオブジェクト
18
ニックネームの登録 224
ニックネーム・オプション 605
フェデレーテッド・システムへの追加
218
へのアクセスの構成 218
無効な WHERE 文節 228
メッセージ 239
ラッパーの登録 220
ラッパー・ライブラリー・ファイル
221

Entrez (続き)
CREATE SERVER ステートメント
604
Nucleotide スキーマ表 235
PubMed スキーマ表 230
SQL スキーマ中のデータ・エレメント
228
EnzymesByPathwayS ユーザー定義関数
構文 543
EnzymesByPathwayT ユーザー定義関数
構文 543
Excel ファイル
アクセスの準備 29
サーバー定義の登録 247
サポートされるバージョン 6
サンプルの照会 248
サンプル・ユーザーのシナリオ 249
説明 243
データ・タイプ、サポートされる 569
ニックネーム、有効なオブジェクト
18
ファイル・アクセス制御モデル 250
フェデレーテッド・システムへの追加
244
ニックネームの登録 247
CREATE NICKNAME ステートメ
ント 247
へのアクセスの構成 244
メッセージ 251
ラッパーの登録 245
ラッパー・ライブラリー・ファイル
246
CREATE NICKNAME 構文 597
CREATE SERVER ステートメント
597
ODBC ラッパーを使用するのアクセス
353
Extended Search
関数テンプレートの登録 269
検索可能データ・ソース 257
サーバー定義の登録 266
サポートされるバージョン 6
サンプルの照会 272
照会のガイドライン 270
照会のカスタマイズ 269
垂直表 261
説明 257
データ・タイプ、サポートされる 569
ニックネーム、有効なオブジェクト
18
ニックネームの登録 268
汎用照会言語 274
フェデレーテッド・システムへの追加
264
複数のニックネームの作成 260
へのアクセスの構成 264

Extended Search (続き)

- マップ・フィールド 261
- メッセージ 276
- ユーザー・マッピングの登録 267
- ラッパーの登録 265
- ラッパー・ライブラリー・ファイル 266
- CREATE FUNCTION ステートメント 269, 604
- CREATE NICKNAME 構文 601
- CREATE NICKNAME ステートメント 268
- CREATE SERVER ステートメント 599
- CREATE USER MAPPING ステートメント 267, 601
- CREATE WRAPPER ステートメント 599

F

FaultQueue

- ビジネス・オブジェクト・アプリケーション構成 139

FEDERATED パラメーター設定 49

FOLD_ID サーバー・オプション

- 値の大文字小文字が区別される 26
- 有効な設定値 629
- Informix データ・ソースでの設定 322

FOLD_PW サーバー・オプション

- 値の大文字小文字が区別される 26
- 有効な設定値 629
- Informix データ・ソースでの設定 322

G

GenesByMotifsS ユーザー定義関数構文 564

GenesByPathwayS ユーザー定義関数構文 540

GenesByPathwayT ユーザー定義関数構文 541

GeneWise 516, 518

GeneWise UDF ライブラリーへのリンク 516

Genome データベース 217

H

H2X_PATH、HMMER デーモン 286

HMMER ニックネームの固定列 297

HMMER (続き)

- ラッパー・ライブラリー・ファイル 294
- CREATE SERVER ステートメントの例 295
- HMMER データ・ソース
- サーバー定義の登録 294
- サポートされるバージョン 6
- サンプルの照会 302
- 照会 304
- 説明 281
- データ・タイプ、サポートされる 569
- デーモン
- 開始 290
- 構成 286
- デーモン構成ファイルの例 290
- ニックネーム、有効なオブジェクト 18
- ニックネームの登録 296
- フェデレーテッド・システムへの追加 284
- へのアクセスの構成 284
- メッセージ 305
- ユーティリティのリスト 281
- ラッパーの登録 293
- CREATE NICKNAME ステートメントの例 301
- hmmpfam 実行可能ファイル 285
- HMMERPFAM_PATH、HMMER デーモン 286
- HMMER_OUT_DIR_PATH HMMER デーモン 286
- hmmpfam 実行可能ファイル
- インストール済みバージョンの検査 285
- 説明 281

I

IFILE サーバー・オプション有効な設定値 629

IGNORE_UDT サーバー・オプション有効な設定値 629

Informix

- 環境変数、設定 309
- クライアント構成ファイルのセットアップ 308
- クライアント構成ファイルのテスト 308
- サーバー接続のテスト 319
- サーバー定義の登録 315
- サポートされるバージョン 6
- チューニング 322
- デフォルトの逆方向タイプ・マッピング 685

Informix (続き)

- デフォルトの順方向タイプ・マッピング 669
 - デフォルトのラッパー名 14
 - ニックネーム、有効なオブジェクト 18
 - ニックネームの登録 320
 - フェデレーテッド・システムへの追加 307
 - へのアクセスの構成 307
 - ユーザー・マッピングの登録 318
 - 有効なサーバー・タイプ 623
 - ラッパーの登録 313
 - ラッパー・ライブラリー・ファイル名 314
 - CREATE NICKNAME ステートメントの例 321
 - CREATE SERVER ステートメント 316
 - CREATE USER MAPPING ステートメント 318
 - INFORMIX_LOCK_MODE サーバー・オプション有効な設定値 629
 - INITIAL_INSTS 関数マッピング・オプション有効な設定値 697
 - INITIAL_IOS 関数マッピング・オプション有効な設定値 697
 - INSTS_PER_ARGBYTE 関数マッピング・オプション有効な設定値 697
 - INSTS_PER_INVOC 関数マッピング・オプション有効な設定値 697
 - IOS_PER_ARGBYTE 関数マッピング・オプション有効な設定値 697
 - IOS_PER_INVOC 関数マッピング・オプション有効な設定値 697
 - IO_RATIO サーバー・オプション有効な設定値 629
 - IUD_APP_SVPT_ENFORCE サーバー・オプション有効な設定値 629
- ## K
- KEGG ユーザー定義関数
 - 概要 535
 - 関数のリスト 536
 - 除去 565
 - 登録 538
 - 引き数記述 537

KEGG ユーザー定義関数 (続き)
SSDB に戻された列 549

L

link-edit
djaxlink の参照 43
LOGIN_TIMEOUT サーバー・オプション
有効な設定値 629
LSBarCode ユーザー定義関数
構文 518
例 519
LSDefineParse ユーザー定義関数
構文 506
例 511
LSGeneWise ユーザー定義関数 516, 518
ライブラリー 495
LSMultiMatch ユーザー定義関数
構文 520
例 521
LSMultiMatch3 ユーザー定義関数
構文 522
例 522
LSNuc2Pep ユーザー定義関数
構文 527
例 528
LSPatternMatch ユーザー定義関数
構文 512
例 512
LSPep2AmbNuc ユーザー定義関数
エラー・メッセージ 503
構文 500
例 501
LSPep2ProbNuc ユーザー定義関数
エラー・メッセージ 505
構文 503
コドン度数表の形式 531
例 504
LSPrositePattern ユーザー定義関数
構文 514
例 514
LSRevComp ユーザー定義関数
構文 524
例 524
LSRevNuc ユーザー定義関数
構文 525
例 526
LSRevPep ユーザー定義関数
構文 526, 529
例 527, 530

M

MAX_PENDING_REQUESTS
BLAST デモン 113

MAX_PENDING_REQUESTS (続き)
HMMER デモン 286

Microsoft Excel

Excel ファイルを参照 6

Microsoft SQL Server

環境変数、設定 328
サーバー接続のテスト 336
サーバー定義の登録 332
サポートされるバージョン 6
チューニング 339
デフォルトの逆方向タイプ・マッピング
685
デフォルトの順方向タイプ・マッピング
669
デフォルトのラッパー名 14
トラブルシューティング 339
ニックネーム、有効なオブジェクト
18
ニックネームの登録 338
フェデレーテッド・サーバーの準備
326
フェデレーテッド・システムへの追加
325
へのアクセスの構成 325
ユーザー・マッピングの登録 335
有効なサーバー・タイプ 623
ラッパーの登録 330
ラッパー・ライブラリー・ファイル名
331
CREATE NICKNAME ステートメント
の例 339
CREATE SERVER ステートメント
333
CREATE USER MAPPING ステートメント
の例 335
MODULE ラッパー・オプション
有効な設定値 621
Motif UDF
LSBarCode 518
LSMultiMatch 520
LSMultiMatch3 522

N

NET8 ラッパー
ライブラリー・ファイル名、
Oracle 372
NLS (ナショナル・ランゲージ・サポ
ート) 51
NODE サーバー・オプション、有効な設
定値 629
Nucleotide
スキーマ表 235
データベース 217
NUMERIC_STRING 列オプション
有効な設定値 661

NUMERIC_STRING 列オプション (続き)
例 581

O

Object Discovery Agent
ビジネス・アプリケーション 137
ビジネス・アプリケーション・ラッパ
ー 139
ODBC
サーバー接続のテスト 350
サーバー定義の登録 346
サポートされるバージョン 6
チューニング 357
デフォルトの順方向タイプ・マッピング
669
デフォルトのラッパー名 14
トラブルシューティング 357
ニックネーム、有効なオブジェクト
18
フェデレーテッド・サーバーの準備
343
フェデレーテッド・システムへの追加
341
ニックネームの登録 351
へのアクセスの構成 341
ユーザー・マッピングの登録 348
有効なサーバー・タイプ 623
ラッパーの登録 344
ラッパー・ライブラリー・ファイル
345
CREATE NICKNAME ステートメント
の例 353
CREATE SERVER ステートメント
347
CREATE USER MAPPING ステートメ
ント 349
CREATE WRAPPER ステートメント
345
ODBC ラッパー
Excel ファイルへのアクセス 353
OLE DB
サーバー定義の登録 361
サポートされるバージョン 6
デフォルトのラッパー名 14
フェデレーテッド・システムへの追加
概要 359
へのアクセスの構成 359
ユーザー・マッピングの登録 363
有効なサーバー・タイプ 623
ラッパーの登録 360
ラッパー・ライブラリー・ファイル
361
CREATE SERVER ステートメント
362

OLE DB (続き)

CREATE USER MAPPING ステートメント 364

OMIM データベース 217

Oracle

環境変数、設定 366

クライアント構成ファイルのセットアップ 370

クライアント構成ファイルのテスト 370

サーバー接続のテスト 377

サーバー定義の登録 373

チューニング 379

デフォルトの逆方向タイプ・マッピング 685

デフォルトの順方向タイプ・マッピング 669

デフォルトのラッパー名 14

トラブルシューティング 379

ニックネーム、有効なオブジェクト 18

フェデレーテッド・システムへの追加 365

ニックネームの登録 378

へのアクセスの構成 365

ユーザー・マッピングの登録 375

ライブラリー名 372

ラッパーの登録 371

ラッパー名 372

例、CREATE USER MAPPING ステートメント 375

CREATE NICKNAME ステートメントの例 378

CREATE SERVER ステートメント 374

P

PACKET_SIZE サーバー・オプション
有効な設定値 629

ParalogsByGeneS ユーザー定義関数
構文 558

ParalogsByGeneT ユーザー定義関数
構文 559

PASSWORD サーバー・オプション
有効な設定値 629

Pathway データベース
関数のリスト 536

Pathway データベース UDF
CompoundsByPathwyS 541
CompoundsByPathwyT 542
EnzymesByPathwyS 543
EnzymesByPathwyT 543
GenesByPathwyS 540
GenesByPathwyT 541
PathwysByCompndsS 546

Pathway データベース UDF (続き)

PathwysByCompndsT ユーザー定義関数
546

PathwysByEnzymesS 547

PathwysByEnzymesT 548

PathwysByGenesS 544

PathwysByGenesT 545

PathwysByCompndsS ユーザー定義関数
構文 546

PathwysByCompndsT ユーザー定義関数
構文 546

PathwysByEnzymesS ユーザー定義関数
構文 547

PathwysByEnzymesT ユーザー定義関数
構文 548

PathwysByGenesS ユーザー定義関数
構文 544

PathwysByGenesT ユーザー定義関数
構文 545

PCRE ライブラリー・パッケージ 515

PeopleSoft
ビジネス・オブジェクト 135

PERCENT_ARGBYTES 関数マッピング・
オプション
有効な設定値 697

PLAN_HINTS サーバー・オプション
有効な設定値 629

PubMed スキーマ表 230

PubMed データベース 217

PUSHDOWN サーバー・オプション
有効な設定値 629

Q

Q_SEQ_DIR_PATH

BLAST デモン 113

HMMER デモン 286

R

REMOTE_AUTHID ユーザー・オプション
有効な設定値 647

REMOTE_DOMAIN ユーザー・オプション
有効な設定値 647

REMOTE_NAME 関数マッピング・オプション
有効な設定値 697

REMOTE_PASSWORD ユーザー・オプション
有効な設定値 647

RequestQueue

ビジネス・オブジェクト・アプリケーション構成 139

ResponseQueue

ビジネス・オブジェクト・アプリケーション構成 139

RevBestNbrsByGeneS ユーザー定義関数
構文 556

RevBestNbrsByGeneT ユーザー定義関数
構文 557

RUNPFAM_PATH

HMMER デモン 286

S

SAP

ビジネス・オブジェクト 135

ScoreBetweenGenesS ユーザー定義関数
構文 563

Sequence Similarity データベース
関数のリスト 536

戻された列 549

Sequence Similarity データベース UDF

AllNbrsByGeneS 550

AllNbrsByGeneT 551

BestHmlgsByGenesS 560

BestHmlgsByGenesT 560

BestNbrsByGeneS 554

BestNbrsByGeneT 555

BstBstHmlgByGenesS 561

BstBstHmlgByGenesT 562

BstBstNbrsByGeneS 552

BstBstNbrsByGeneT 553

DefinitionsByGeneS 563

GenesByMotifsS 564

ParalogsByGeneS 558

ParalogsByGeneT 559

RevBestNbrsByGeneS 556

RevBestNbrsByGeneT 557

ScoreBetweenGenesS 563

SET SERVER OPTION ステートメント
一時的にオプションを設定する 15

Siebel

ビジネス・オブジェクト 135

SQL コンパイラー

フェデレーテッド・システム内の 10

SQL ダイアレクト

説明 11

SQLNET ラッパー

ライブラリー・ファイル名 372

Sybase

環境変数、設定 382

クライアント構成ファイルのセットアップ 384

クライアント構成ファイルのテスト 384

サーバー接続のテスト 392

サーバー定義の登録 387

サポートされるバージョン 6

Sybase (続き)

- チューニング 395
 - デフォルトの逆方向タイプ・マッピング 685
 - デフォルトの順方向タイプ・マッピング 669
 - デフォルトのラッパー名 14
 - トラブルシューティング 395
 - ニックネーム、有効なオブジェクト 18
 - フェデレーテッド・システムへの追加 381
 - ニックネームの登録 393
 - へのアクセスの構成 381
 - ユーザー・マッピングの登録 391
 - 有効なサーバー・タイプ 623
 - ラッパーの登録 386
 - ラッパー・ライブラリー・ファイル名 387
 - 例、CREATE USER MAPPING ステートメント 391
 - CREATE NICKNAME ステートメントの例 394
 - CREATE SERVER ステートメント 389
- SYSCAT カタログ・ビュー 617
- SYSSTAT カタログ・ビュー 617

T

Teradata

- アクセス・ロギング 424
- 環境変数、設定 412
- サーバー接続のテスト 410, 420
- サーバー定義の登録 416
- チューニング 424
- デフォルトの逆方向タイプ・マッピング 685
- デフォルトの順方向タイプ・マッピング 669
- デフォルトのラッパー名 14
- トラブルシューティング 424
- ニックネーム、有効なオブジェクト 18
- フェデレーテッド・サーバーでのニックネームの概要 422
- フェデレーテッド・システムへの追加 409
 - ニックネームの登録 422
 - ランタイム・リンクの検査 (AIX) 411
- へのアクセスの構成 409
- ユーザー・マッピングの登録 419
- 有効なサーバー・タイプ 623
- ラッパーの登録 415

Teradata (続き)

- ラッパー・ライブラリー・ファイル名 416
- ランタイム・リンクの使用可能化 (AIX) 411, 424
- CREATE NICKNAME ステートメントの例 423
- CREATE SERVER ステートメント 418
- CREATE USER MAPPING ステートメント 420
- TIMEFORMAT サーバー・オプション 有効な設定値 629
- TIMEOUT サーバー・オプション 有効な設定値 629
- TIMESTAMPFORMAT サーバー・オプション 有効な設定値 629
- TurboBlast 128

U

- UDF の定義行構文解析
 - 概要 506
 - 説明 122
- LSDefineParse 506
- UDF の変換
 - LSNuc2Pep 527
 - LSTransAllFrames 529
- Unicode 55

V

- VARCHAR_NO_TRAILING_ BLANKS サーバー・オプション 有効な設定値 629
- VARCHAR_NO_TRAILING_ BLANKS 列オプション 有効な設定値 661
- 例 581

W

Web サービス

- サーバー定義の登録 435
- サンプルの照会 456
- データ・タイプ、サポートされる 569
- トラブルシューティング 460
- ニックネーム 427, 446
- ニックネームの登録 436
- フェデレーテッド・システムへの追加 433
- へのアクセスの構成 433
- ラッパー・ライブラリー・ファイル 435

Web サービス (続き)

- 例 169, 452
- Web サービス記述言語 (WSDL)
 - ニックネーム 427
 - ニックネームへのマッピング 154, 443
- Web サービス・プロバイダー トラブルシューティング 460
- Web サービス・ラッパー 作成 434
- WebSphere Business Integration 構成 137
- WebSphere Business Integration アダプター
 - インストール 139
- WebSphere Business Integration ラッパー 135
 - サーバー定義 145
 - 作成 143
 - データ・タイプ、サポートされる 569
 - ニックネームの登録 146
 - ラッパー・ライブラリー・ファイル 144
 - 例 156, 169, 452
- WebSphere MQ
 - メッセージ・キュー 139

X

XML

- コスト・モデル機能 473, 480
- サーバー定義の登録 469
- サポートされるバージョン 6
- サンプルの照会 481
- 説明 463
- ソース文書およびニックネーム 471
- データ・タイプ、サポートされる 569
- ニックネーム、有効なオブジェクト 18
- 非ルート・ニックネームのフェデレーテッド・ビューの作成 479
- フェデレーテッド・システムへの追加 466
 - ニックネームの登録 471, 474
 - CREATE NICKNAME ステートメント 474
- へのアクセスの構成 466
- メッセージ 482
- ラッパーの登録 467
- ラッパー・ライブラリー・ファイル 468
- CREATE NICKNAME 構文 610
- CREATE NICKNAME ステートメントの例 474

XML スキーマ

ニックネームへのマッピング 154,
443

ビジネス・アプリケーション 137

XPath

ニックネームとの使用 154, 443

IBM と連絡を取る

技術上の問題がある場合は、お客様サポートにご連絡ください。

製品情報

DB2 Information Integrator についての情報は、電話または Web から入手することができます。

米国にお住まいの場合は、以下のいずれかの番号にお問い合わせください。

- 製品の注文または一般情報の入手: 1-800-IBM-CALL (1-800-426-2255)
- 資料の注文: 1-800-879-2755

Web 上で <http://www.ibm.com/software/data/integration/db2ii/support.html> にアクセスします。このサイトには、最新のテクニカル・ライブラリーに関する情報、資料の注文、クライアントのダウンロード、ニュースグループ、フィックスパック、ニュース、および Web リソースのリンクが含まれています。

お住まいの国や地域の IBM 事務所の所在地を調べる場合は、Web 上で IBM Directory of Worldwide Contacts (www.ibm.com/planetwide) を参照してください。

資料についてのコメント

お客様のフィードバックは IBM が良質な情報を提供する助けになります。この資料や他の DB2 Information Integrator の資料についてのコメントをお送りください。コメントの送付には、以下のいずれかの方法を利用することができます。

- www.ibm.com/software/data/rcf で、オンラインの読者コメント・フォームを使用して送信する。
- 電子メール (E メール) で comments@us.ibm.com に送信する。お送りいただく情報には、製品の名前、製品のバージョン番号、および資料の名前と部品番号 (該当する場合) を必ず含めてください。特定の本文についてコメントする場合は、本文の位置 (たとえば、タイトル、表の番号、またはページ番号) を含めてください。



Printed in Japan